USING THE RELATIONAL MODEL TO CAPTURE TOPOLOGICAL INFORMATION OF SPACES

PATRICK ERIK BRADLEY AND NORBERT PAUL

Abstract. Motivated by research on how topology may be a helpful foundation for building information modeling (BIM), a relational database version of the notions of chain complex and chain complex morphism is defined and used for storing cw-complexes and their morphisms, hence instances of building projects and different views upon them, into relational databases. In many cases, this can be done without loss of topological information. The equivalence of categories between sets with binary relations and Alexandrov spaces is proven and used to incorporate the relational complexes into the more general setting of topological databases. For the latter, a topological version of a relational query language is defined by transferring the usual relational algebra operators into topological constructions. In the end, it is proven that such a topological version of relational algebra in general must be able to compute the transitive closure of a relation.

1. Introduction

In building information modeling (BIM), there are many important topological properties of architectural buildings which must be representable by such a model. If, for example, two walls of a building are adjacent in a common edge, it must be possible to retrieve this adjacency information from the corresponding data model in order to be able to compute important planning information like heat flows or structural performance. In facility management, as another example, one often wants to know by which paths a room is accessible—such information enables sophisticated access control planning.

Spatial modelling, such as product data modelling, volume modelling or geographic information systems, is always confronted with topological information. Hence, the storage and retrieval of topological information of spaces is an important issue in these domains. In geographic information systems topologies are stored using graphs. The data structures used there are very similar to those used by those volume modelers which are based on so-called boundary representation (B-rep) techniques [1]. On the other hand, in algebraic topology the notion of complex is a natural generalisation of B-rep modeling to higher dimension.

An important issue in the encoding of spaces is, how much topological information is kept in the data model. It is well-known, however, that already for graphs the chain complex corresponding to some given graph does not recover its isomorphism class. This means that there is a loss in topological information. There are different kinds of remedies for this problem: one well known such is implemented as DIME (Dual Independent Map Encoding) which works well for planar graphs and which is frequently used in geoinformation systems. Another approach is by

Date: April 25, 2008.

storing the local observation structures around vertices, a method which allows generalisation to higher dimension [2].

The usual way of dealing with building information in architecture is to read, write and exchange files. As an alternative, we propose the usage of a common database. Traditionally, this is accomplished by a relational database server. However, we advocate to extend their functionality to handling information of at least three-dimensionsional spaces. Our aim here is to lay the theoretical foundation upon which general spatial information systems can be built.

There exists a zoo of data structures for spatial modeling, like the above mentioned DIME, the Winged Edge structure, or the topology-resources in ISO STEP 10303-42, which are mostly based on cell complexes. In this article, we lay a formal foundation for such data structures. In particular, we give a generalisation of chain complex which encaptures more topological information than the usual chain complexes by using partial matrices as boundary operators.

As our motivation stems from building information modeling, we state that architectural design amounts to partitioning space in such a way that one obtains a finite cw-complex. Of course, this depends on postulating that architectural design is making explicit a finite partitioning of space in the first place.

From the point of view we take, it is very natural to use a categorial language, and we keep this throughout the whole article. The reason is that not only spaces themselves are of interest, but also their considerations at various levels of detail or their being subject to modifications. This necessitates the modelling of continuous maps between topological spaces.

In fact, we observe here that any set with a binary relation on it (i.e. a simple graph) is essentially a so-called *Alexandrov* topological space, and every Alexandrov space has such a graph. These simple graphs are called by us *topological datatypes* or *topological databases*, as we have their obvious database implementations in mind. We prove that the category of topological datatypes is equivalent to the category of Alexandrov spaces.

Cw-complexes fit into the picture, because there is a functor from these to topological datatypes. We describe the essential image of this functor in terms of the graphs which the corresponding relations give.

Our main result for spatial modeling is a surprisingly simple relational database scheme for chain complexes of arbitrary dimension in which this dimension is given by an attribute only and which is applicable for all finite topological spaces. Another result in database theory is the fact that for computing various topological operations and for checking maps on continuity, in general the transitive closure of relations needs to be computed.

After Section 2 which briefly introduces the notions from topology used in this article, Section 3 discusses the topological aspects of architecture and postulates the importance of cw-complexes in architectural design, of course from a BIM perspective.

In Section 4, we introduce the notion of *relational complex* which allows marked chain complexes to be transferred into a relational database[1]. It seems natural, from the point of view of relational databases, to use partial matrices for representing the boundary operator. As a consequence, more topological information can be

[1]The step therefrom to an object oriented implementation is trivial.

recovered than with usual chain complexes, if the complex comes from a finite cw-complex. The corresponding category is called **DChainComp**.

In Section 5, we prove the equivalence of the category **DTop**, consisting of sets with a binary relation on it, and the category **Alex** whose objects are the Alexandrov topological spaces with continuous maps as morphisms. This slightly generalises a result by Alexandrov in [3], where he proved the equivalence in the special case of sets having a partial ordering relation and a special class of T_0-spaces. Even if Alexandrov at his time had not the language of categories and functors at hand, we feel we should attribute our result of this section to him, because (the translation into the categorial language) of his proof carries over in a straightforward manner to the more general situation. Any combinatorial cw-complex yields in a natural way a set with a directed acyclic graph ordering on it, and is thus covered by Alexandrov's result.

In Section 6, we perform some topological constructions in **DTop** by giving relations which generate the constructed topologies. In fact, our constructions are the initial and final topologies and special cases thereof. In particular the standard operators of relational algebra are such special cases. Finally, we prove the necessity of the transitive closure computation in any query language which decides continuity of maps between Alexandrov spaces, or which computes any initial topology.

A part of the results from this article has been announced without proofs in [4]. The ideas of viewing architecture as finite topological space and of extending the relational model by topology are attributed to the second author and are expanded in his dissertation [5]. A first prototypical implementation is discussed in [6].

2. Topological preliminaries

We assume familiarity with the basic notions from set-theoretic and algebraic topology as well as some knowledge of the language of categories and functors. The aim of this section is to settle definitions and notations of objects treated in this article.

A *chain complex* will be understood as an infinite sequence of $\mathbb{Z}$-modules C_n

$$\mathcal{C}\colon \ \ldots \xrightarrow{\partial_{n+1}} C_n \xrightarrow{\partial_n} C_{n-1} \xrightarrow{\partial_{n-1}} \cdots$$

with the usual property $\partial_{n-1} \circ \partial_n = 0$ for all $n \in \mathbb{Z}$. We will also consider it in its equivalent form as a pair $(C = \bigoplus C_n, \partial)$ with C a graded abelian group and the boundary operator $\partial\colon C \to C$ having the properties

$$\partial(C_n) \subseteq C_{n-1} \quad \text{and} \quad \partial^2 = 0.$$

All our complexes will satisfy $C_n = 0$ for $n < 0$ and most complexes will be finite. The latter means that all but finitely many of the C_n will be the zero module 0. All modules occurring in our complexes will be finitely generated free abelian groups. These have the advantage of always having a basis. By fixing a basis for a given module, we obtain a *marked chain complex*. Together with the chain maps (also called morphisms of chain complexes), these form the category **MChainComp**. Recall that chain maps $f\colon (C, \partial) \to (C', \partial')$ are linear maps which respect the boundary operators: $f \circ \partial = \partial' \circ f$.

Of particular interest in this article are cw-complexes with finitely many cells. We will call such a *finite cw-complex* or, by abuse of language, just *cw-complex*. Recall that a *cellular map* of cw-complexes takes n-cells to m-cells with $m \leq n$.

The category of cw-complexes will be denoted by **CW**. There is the well-known functor **CW** $\to$ **MChainComp** which takes each cw-complex X to the marked chain complex $\mathcal{C}(X)$ whose basis is the set B of cells of X. The boundary operator ∂ is defined on n-cells $b \in B_n$ as

$$\partial(b) = \sum_{c \in B_{n-1}} [b : c] \cdot c,$$

where $[b : c] \in \mathbb{Z}$ is the degree of the map f_{bc} between spheres S^n induced by the map which attaches b to X along its boundary S^n:

$$f_{bc} \colon S^{n-1} \to X^{n-1}/(X^{n-1} \setminus b) \cong S^{n-1},$$

where X^{n-1} is the $n-1$-skeleton of X. A significant portion of this article will discuss a relational version of this functor.

A *combinatorial cw-complex* is the quotient of a cw-complex X by the equivalence relation whose classes are precisely the cells. It is endowed with the quotient topology of the weak topology of X. Together with the continuous maps, they form the category **CombCW**. Allowing only the cellular maps as morphisms, we obtain the subcategory **CombCW**^{c}.

3. Architectural design and complexes

Our initial motivation for our work was the assumption, that the topological spaces arising from architecture can be helpful for the development of building information systems (BIM). These spaces are finite topological spaces obtained by partitioning the real Euclidean space $\mathbb{R}^n$ into finite pieces representing architectural design. We will say that this partition is a *finite representation* of n-space. Note that we do not restrict to the case of dimension three for two reasons: firstly, the theory is sufficiently general to handle any finite dimension; secondly, the space which encaptures all relevant building information is itself usually more than three-dimensional.

An architectural space has some similarities to a cw-complex. It is composed of manifolds instead of cells, but these manifolds still cover a compact subspace of the n-dimensional real space.

Definition 3.1 (Architectural representation)**.** *Let $\sim$ be a finite representation of $\mathbb{R}^n$, and $K \subseteq \mathbb{R}^n$ a compact subset. Let $\sim_K := \sim|_{K \times K}$ be the restriction of $\sim$ to $K \times K$. We say that the pair $(K, \sim_K)$ is an* architectural representation*, if K is a union of equivalence classes for $\sim$, and each equivalence class e for $\sim$ is homeomorphic to an open connected subset of some $\mathbb{R}^q$ for some $q \in \mathbb{N}$. We call e an* architectural element *in $(X, \sim_K)$ and q the* dimension *of e.*

For two architectural representations $(K, \sim)$ and $(S, \approx)$ we call a continuous mapping $f : K \to S$ an (architectural) representation mapping *if the pre-image $f^{-1}([s])$ of each equivalence class $[s] \in S/\approx$ is a union of equivalence classes in $K/\sim$. We then write $f : (K, \sim) \to (S, \approx)$* □

Remark 3.2. It is a well known result from algebraic topology that if U is a subset of $\mathbb{R}^n$ homeomorphic to some open subset of $\mathbb{R}^q$, then necessarily $q \leq n$.

Remark 3.3. A given representation mapping $f \colon (K, \sim) \to (S, \approx)$ induces a continuous mapping

$$f_\approx \colon K/\sim \to S/\approx,\ [k] \mapsto f_\approx([k]) := [f(k)]$$

between the quotient spaces. The composition of two representation mappings is a representation mapping, too, and the identity $\mathrm{id} : (K, \sim) \to (K, \sim)$ is also a representation mapping. Hence, the architectural representations together with the representation mappings form a category which we denote by **ARep**.

An architectural element is a cell with possibly some holes cut out. Such an element may be the facade, a room, a window glass. Later we will become less particular with "homeomorphic" and allow a certain thickness of architectural elements. Then a door qualifies as a 2-dimensional element and its frame has dimension 1.

We postulate the following property of architecture:

Postulate 1. *Architectural design is the making explicit of an architectural representation of* $\mathbb{R}^n$*.*

This postulate still allows finite partitionings which cannot be refined into finite cw-complexes. So we restrict architectural design in a way such that the similarity to cw-complexes will become even stronger.

Postulate 2. *Any architectural representation* $(K, \sim)$ *of an* n*-space made by architectural design in practice has a refinement* $\approx$ *of* $\sim_K$ *such that* $(K, \approx)$ *is a finite cw-complex.*

By a *refinement* of an equivalence relation $\sim$ on K we mean an equivalence relation $\approx$ on K such that

$$x \approx y \implies x \sim y$$

holds true for all $x, y \in K$. Thus, by Remark 3.2 the equivalence classes of the refinement $\approx$ in Postulate 2 are indeed q-cells for q between 0 and n.

The sections to follow are an individual treatment of the two postulated topological aspects of architectural space: set-theoretic versus algebraic.

4. The category of relational schemes for chain complexes

We give a straightforward definition of a chain complex within the context of the relational model. This means that partial matrices become important, because the matrices occurring in the context of marked complexes usually are so-called sparse matrices having lots of zero entries which can be removed to save storage space. We do not want to remove all zero entries, however, because some of them are too useful to be discarded.

4.1. **Partial matrices.**

Here, we review some matter about partial matrices which suits our purposes, and fix some notation.

Recall that a matrix with integer values is a map

$$A \colon I \times J \to \mathbb{Z},\ (i, j) \mapsto \alpha_{ij},$$

where I and J are sets. If I and J are finite of cardinalities m and n, then we have an $m \times n$-matrix.

Definition 4.1 (Partial matrix)**.** *A* partial (integer) matrix $A = (\alpha_{ij})$ *is a partial map*

$$A :\subseteq I \times J \to \mathbb{Z},\ (i, j) \mapsto \alpha_{ij},$$

The maximal subset $U \subseteq I \times J$*, on which the restriction*

$$A|_U \colon U \to \mathbb{Z},\ (i, j) \mapsto \alpha_{ij},$$

is a map, is called the domain *of A, and denoted by $\delta(A)$. If $\delta(A) = I \times J$, then we say that A is a* total *matrix.* □

If a partial matrix $A :\subseteq I \times J \to \mathbb{Z}$ is not defined in $(i,j) \in I \times J$, then we will write $\alpha_{ij} = \texttt{null}$ or $\alpha_{ij} = \uparrow$.

As for ordinary matrices, we say that a partial matrix

$$A :\subseteq I \times J \to \mathbb{Z},\ (i,j) \mapsto \alpha_{ij}$$

is a *partial $m \times n$-matrix*, if $\operatorname{card}(I) = m$ and $\operatorname{card}(J) = n$. The set of partial integer $m \times n$-matrices will be denoted by

$$\mathbb{Z}_{\subseteq}^{m \times n},$$

whereas $\mathbb{Z}^{m \times n}$ stands for the set of all total $m \times n$-matrices.

Also, we extend the matrix product in a natural way to a product of the partial $m \times n$-matrix A and a partial $n \times p$-matrix

$$B :\subseteq J \times K \to \mathbb{Z},\ (j,k) \mapsto \beta_{ij},$$

by summing up only over indices on which both matrices are defined and saying the sum over an empty set is undefined, hence $\sum_{i \in \emptyset} a_i := \uparrow$.

More precisely, $A \cdot B = (\gamma_{ij})$, where

$$\gamma_{ij} = \sum_{\ell\colon (i,\ell) \in \delta(A),\ (\ell,j) \in \delta(B)} \alpha_{i\ell}\beta_{\ell j}. \tag{1}$$

This leads to the following rules for calculating with $\uparrow$:

$$\uparrow + a = a + \uparrow = a \quad \text{for } a \in \mathbb{Z} \cup \{\uparrow\}$$
$$\uparrow \cdot a = a \cdot \uparrow = \uparrow \quad \text{for } a \in \mathbb{Z} \cup \{\uparrow\}$$

Note that these rules differ from the usual calculation rules with `null`-values in relational databases. Usually these `null`-values are supposed to be a placeholder for missing information. We clearly do not advocate a general replacement of these rules by the rules presented above. In our case, however, $\uparrow$ is introduced to actually *store* additional information and these rules are more adequate for this kind of application.

Note also, that the domain of a matrix can be viewed as a relation and that $\delta(A \cdot B) = \delta(A) \circ \delta(B)$ holds: The domain of the partial matrix product is the relational product of the domains of the matrices involved.

Example 4.2 (Unity partial matrix)**.** Let

$$1_n :\subseteq \{1,\dots,n\} \times \{1,\dots,n\} \to \mathbb{Z},$$
$$(i,j) \mapsto \begin{cases} 1, & i = j \\ \uparrow, & \text{otherwise} \end{cases}$$

be the *minimal identity partial $n \times n$-matrix* or *unity partial $n \times n$-matrix*. The multiplication rule for partial matrices yield for $A \in \mathbb{Z}_{\subseteq}^{m \times n}$:

$$A \cdot 1_n = 1_m \cdot A = A. \tag{2}$$

Note that the unity rule (2) does not hold in general, if one replaces in 1_A some $\uparrow$ off the diagonal by 0. Let namely $I \in \mathbb{Z}_{\subseteq}^{m \times m}$ be such a matrix with the extra zero in the place (i,j), and $A = (\alpha_{rs}) \in \mathbb{Z}_{\subseteq}^{m \times n}$ be such that $\alpha_{ik} = \uparrow$ for some k, and $\alpha_{jk} \in \mathbb{Z}$ for $j \neq i$. Then the (i,k)-element of $I \cdot A$ equals $1 \cdot \uparrow + \cdots + 0 \cdot \alpha_{jk} \neq \uparrow$.

Example 4.3 (Partial zero matrix)**.** A *partial zero $m \times n$-matrix* is any matrix in $\mathbb{Z}_{\subseteq}^{m\times n}$ containing only 0 or $\uparrow$ entries. We will use the notation 0 for any partial zero matrix of any size. The particular size is usually clear from the context. However, we will only specify the domain of the partial zero matrix when explicitly needed.

We say that a partial $n \times n$-matrix A is *invertible*, if there is a matrix B such that $A \cdot B = B \cdot A = 1_n$. This means that invertible partial matrices have lots of undefined entries.

Lemma 4.4. *A partial $n\times n$-matrix is invertible if and only if it is defined precisely in one entry per line and column, and those entries are all invertible elements of $\mathbb{Z}$, i.e. either 1 or -1.*

Proof. A matrix A with precisely one entry per row and column clearly has an inverse if and only if those entries are ± 1.

Assume that A has a completely undefined i-th row. Then the entry at (i, i) in $A \cdot B$ is not defined, hence A not invertible. By a similar argument for columns, it follows that every invertible partial matrix has at least one defined entry per column and row.

Assume that A has a row with two distinct entries $\alpha_{ij}, \alpha_{ik} \neq\uparrow$. The determinant of A is defined in the usual way, and there is an adjoint partial matrix $A^{\#}$ such that $A \cdot A^{\#}$ and $A^{\#} \cdot A$ each equal to $\det(A)$ times a unitary partial matrix I. In the invertible case, clearly $I = 1_n$ must hold true. The entries of $A^{\#}$ are, as in the total case, of the form $\det(A'_{\ell k})$, where $A'_{\ell k}$ is the partial $(n-1)\times(n-1)$-matrix obtained by deleting the ℓ-th row and the k-the column of A and shrinking to the smaller size. Let now $\gamma_{i\ell}$ be the (i, ℓ)-entry of $A \cdot A^{\#}$ with $i \neq \ell$. It holds true that

$$\gamma_{i\ell} = \sum_k \alpha_{ik} \cdot \det(A'_{\ell k}) = \alpha_{ij} \det(A'_{\ell j}) + \text{other terms}$$

By assumption, α_{ij} is defined, and so is the (i, k)-entry in $A'_{\ell j}$: it equals α_{ik}. This implies that all rows and columns of $A'_{\ell j}$ contain at least one defined entry. Hence, $\det(A'_{\ell j}) \neq\uparrow$. Consequently, A cannot be invertible. □

4.2. **The category DChainComp.** We define the category **DChainComp** whose objects are the marked chain complexes generalised in the relational context. The category of marked chain complexes will then be found to be equivalent to a full subcategory of **DChainComp**.

Definition 4.5. *A* relational complex *is a pair $\mathfrak{C} = (B, D)$, where B is a finite set partitioned into subsets $B_0, \ldots, B_n$ and D is the graph of a partial matrix $\partial :\subseteq B \times B \to \mathbb{Z}$ satisfying the following conditions:*

1. *∂ is defined on $x \in B_j \times B_i$ only if $j < i$.*
2. *If $x \in B_j \times B_i$ with $j < i - 1$, then $\partial(x)$ is either 0 or $\uparrow$.*
3. *$\partial^2 = 0$.*

∂ is called the relational boundary*, and the restriction of ∂ to $B \times B_i$ is called the* i-th relational boundary *of $\mathfrak{C}$. The latter will be denoted by ∂_i.* □

Note that in (iii), the term 0 denotes the partial zero matrix from Example 4.3.

Definition 4.6. *Let $\mathfrak{C} = (B, D)$ and $\mathfrak{C}' = (B', D')$ be relational complexes. A* morphism *$\mathfrak{C} \to \mathfrak{C}'$ of relational complexes is the graph of a partial matrix $m :\subseteq B' \times B \to \mathbb{Z}$ satisfying the conditions:*

(1) *If $i \neq j$ and $x \in B'_j \times B_i$, then $m(x) \in \{0, \uparrow\}$.*
(2) $m \cdot \partial = \partial' \cdot m$.

The composition $M' \circ M$ of M with the morphism $M' \colon \mathfrak{C}' \to \mathfrak{C}''$ is defined as the graph of the partial matrix $m' \cdot m :\subseteq B'' \times B \to \mathbb{Z}$. The relational complexes together with their morphisms form the category **DChainComp**. □

The axioms for a category are readily verified for **DChainComp**. Note that the unity morphism $1_{\mathfrak{C}} \in \mathrm{Hom}_{\mathbf{DChainComp}}(\mathfrak{C}, \mathfrak{C})$ is the graph of the minimal unity partial matrix from Example 4.2. In other words, $1_{\mathfrak{C}}$ is the diagonal in $B \times B$.

Remark 4.7. *From the theory of chain complexes, one would expect morphisms of relational complexes to satisfy the relations*

$$m_{i-1} \cdot \delta_i = \delta'_i \cdot m_i, \tag{3}$$

where $m_i :\subseteq B'_i \times B_i$, $\delta_i :\subseteq B_{i-1} \times B_i$ and $\delta'_i :\subseteq B'_{i-1} \times B'_i$ are the restrictions of m, ∂ and ∂', respectively. Namely, these are the translations into the relational setting of the required commuting diagram

$$\begin{array}{ccccccc} \cdots & \longrightarrow & C_i & \xrightarrow{\delta_i} & C_{i-1} & \longrightarrow & \cdots \\ & & \downarrow{\scriptstyle m_i} & & \downarrow{\scriptstyle m_{i-1}} & & \\ \cdots & \longrightarrow & C'_i & \xrightarrow[\delta'_i]{} & C'_{i-1} & \longrightarrow & \cdots \end{array}$$

for morphisms of chain complexes. The relations (3) are indeed satisfied, as the following proof reveals.

Proof. Denote $\alpha_{b'b}$ an entry of the matrix on the left hand side of (3), and $\beta_{b'b}$ the corresponding entry on the right hand side. The indices are given as $b' \in B'_{i-1}$ and $b \in B_i$. Then it holds true that

$$\begin{aligned} \alpha_{b'b} &= \sum_{c \in B_{i-1}} m_{i-1}(b', c)\delta_i(c, b) = \sum_{c \in B_{i-1}} m(b', c)\partial(c, b) \\ &\overset{(*)}{=} \sum_{c \in B} m(b', c)\partial(c, b) \overset{(**)}{=} \sum_{c' \in B'} \partial'(b', c')m(c', b) \\ &\overset{(***)}{=} \sum_{c' \in B'_i} \partial'(b', c')m(c', b) = \sum_{c' \in B'_i} \delta'_i(b', c')m_i(c', b) \\ &= \beta_{b'b}. \end{aligned}$$

The equations $(*)$ and $(***)$ follow from property (i), and $(**)$ from property (ii) of Definition 4.6. □

A relational complex is said to be *total*, if its relational boundary is a total matrix. A morphism $M \colon \mathfrak{C} \to \mathfrak{C}'$ of relational complexes is called *total*, if the underlying partial matrix $m :\subseteq B' \times B \to \mathbb{Z}$ is total. Clearly, total relational complexes with total morphisms form a category $\mathbf{DChainComp}_{\mathrm{tot}}$. However, that category is *not* a subcategory of **DChainComp**, because the unit morphisms do not coincide: in the total case, it is given by the total unity matrix, whereas in **DChainComp**, it is given by the minimal partial unity matrix.

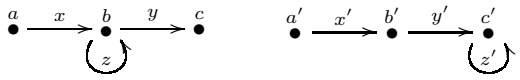


FIGURE 1

Δ_L		
Vertex	Edge	α
a	x	-1
b	x	1
b	y	-1
c	y	1
b	z	0

Δ_R		
Vertex	Edge	α
a	x	-1
b	x	1
b	y	-1
c	y	1
c	z	0

TABLE 1

4.3. **Topological Information in DChainComp.** In this subsection, we will argue how the usage of partial matrices in the category **DChainComp** maintains some topological information which gets lost when making marked chain complexes from cw-complexes.

Example 4.8 (Isomorphic Complexes of Graphs)**.** The two graphs from Figure 1 are not isomorphic but they have the same marked chain complex and therefore the same total relational complex $(\text{Edge}, \text{Vertex}, \Delta)$ where $\text{Edge} = \{x, y, z\}$ is the set of lines, $\text{Vertex} = \{a, b, c\}$ the set of vertices.

The boundary Δ is in both cases the incidence matrix (after identifying a with a', x with x' etc.):

$$\Delta = \begin{pmatrix} \Delta_{a,x} & \Delta_{a,y} & \Delta_{a,z} \\ \Delta_{b,x} & \Delta_{b,y} & \Delta_{b,z} \\ \Delta_{c,x} & \Delta_{c,y} & \Delta_{c,z} \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

However, by removing zeros, we can define different partial matrices Δ_L for the left graph and Δ_R for the right graph:

$$\Delta_L = \begin{pmatrix} -1 & \uparrow & \uparrow \\ 1 & -1 & 0 \\ \uparrow & 1 & \uparrow \end{pmatrix}, \qquad \Delta_R = \begin{pmatrix} -1 & \uparrow & \uparrow \\ 1 & -1 & \uparrow \\ \uparrow & 1 & 0 \end{pmatrix}.$$

Corresponding relational database tables for Δ_L and Δ_R are given in Table 1. Δ_L is defined in (b, z), meaning that $b \in \overline{\{z\}}$. The value 0 of Δ_L in (b, z) means that z is a loop. Δ_R, however, is undefined in (b', z'), meaning that in the right graph we have $b' \notin \overline{\{z'\}}$.

Using Lemma 4.4, it is easy to verify that the two relational complexes defined in this way are not isomorphic.

Example 4.8 contains a special case of a relational encoding of cw-complexes. Namely, let $\mathbb{X}$ be a combinatorial cw-complex with set of cells B. Then (assume that $b \in B_j$ and $c \in B_i$)

$$\partial :\subseteq B \times B \to \mathbb{Z},\ (b, c) \mapsto \begin{cases} [c : b], & b \in \overline{\{c\}} \wedge j = i - 1 \\ 0, & b \in \overline{\{c\}} \wedge j < i - 1 \\ \uparrow, & \text{otherwise} \end{cases}$$

defines the boundary of a relational complex $\mathfrak{X}(\mathbb{X})$.

Definition 4.9. $\mathfrak{C}(\mathbb{X})$ *is called the* relational complex *associated to* $\mathbb{X}$. □

Example 4.10 (Loss of topological information)**.** Our handling of zeros does not recover all topological information, however. This is due to the fact that the boundary of an n-cell c in a relational complex is given by an *unordered* list of $(n-1)$-cells. However, different arrangements of the cells in the boundary of c can lead to non-isomorphic cw-complexes. An example cw-complex X is given by [7, Example 2.38] with non-trivial finite fundamental group

$$\pi_1(X) = \langle a, b \mid a^5 b^{-3} = b^3 (ab)^{-2} = 1 \rangle.$$

A slight rearrangement of 1-cells yields a cw-complex X' having the same relational complex, but with fundamental group

$$\pi_1(X') = \langle a, b \mid a^5 b^{-3} = a^{-2} b = 1 \rangle.$$

The latter is the trivial group, as can be easily calculated.

4.4. **Implementation of relational complexes.** This section briefly describes what an example implementation of relational complexes could look like in Java as described in [6]. Of course other programming languages will do, too. In particular a translation into LISP using CLOS would be straightforward.

4.4.1. *Partial Linear Algebra in Java.* A free $\mathbb{Z}$-module with basis B can be considered as the set $\mathbb{Z}^B$ of the mappings $f\colon B \to \mathbb{Z}$ with addition $f+g : B \to \mathbb{Z}$, $b \mapsto f(b)+g(b)$ and scalar multiplication $z \cdot f : B \to \mathbb{Z}$, $b \mapsto z \cdot f(b)$. An $A \times B$-matrix is nothing else than an element of the free $\mathbb{Z}$-module $\mathbb{Z}^{A\times B}$. The set of partial mappings $h :\subseteq B \to \mathbb{Z}$ is then denoted by $\mathbb{Z}^B_{\subseteq}$ and called the *partial* $\mathbb{Z}$-module with basis B.

Of course we approximate $\mathbb{Z}$ by the Java type `Integer`. We do not use the primitive type `int`, because variables of type `Integer` can have a value `null` which expresses $\uparrow$ in a natural way. Also these objects can be easily stored using the Java Collections Framework. Then a partial $\mathbb{Z}$-module with basis B of type `Set<T>`[2] can be declared by using an appropriate extension of the interface `Map<T,Integer>`. By a simple helper class `Pair<A,B>`, matrices are declared by the following extensions of `Map<Pair<A,B>,Integer>`:

```
IntMap<T> extends Map<T,Integer>
IntMatrix<A,B> extends IntMap<Pair<A,B>>.
```

The linear algebra operations addition, and the various `FOO`-products like scalar multiplication, direct multiplication, inner product and matrix product can be defined in the usual manner as members of these interfaces, such as

```
IntMap<T> IntMap.add(IntMap<T> other)
```

or by using a separate operators class like `IntModule<T>` which operates on such elements by methods like

```
IntMap<T> IntModule.add( IntMap<T> a, IntMap<T> b )
```

sharing some similarity to CLOS' generic functions. Anyhow, the operators must consistently handle undefined entries: If some `m != null` is of type `IntMap<T>`, hence `m` $:\subseteq B \to \mathbb{Z}$, then the property `m`$(b) = \uparrow$ is equivalent to `m.value(b)==null`. We do not permit explicit mappings to `null`, hence `map.containsKey(b)` must then be `false`. In this case for any other `IntMap<T> h` the following must hold:

[2]We assume all necessary package imports like `java.util.*;`.

```
m.value(b) + h.value(b) == h.value(b)&& m.value(b) != 0
```

where `a==b` is shorthand for Java's verbose equality

```
a==null ? b==null : a.equals(b)
```

for reference types and `a!=b` is its negation. This addition, however, would then throw a `NullPointerException`. Hence a utility class `Integers` has to provide methods like `add`, `minus` and `times` which operate on `Integer` references and implement the operations according to the calculation rules in $\mathbb{Z} \cup \{\uparrow\}$ as presented above. These methods must always be used instead of the standard operators. Note that the above plus sign is a shorthand for the method `add`. It is mainly these rules and the additional `null`, why we use the Java type `Integer` instead of the primitive type `int`.

These classes and interfaces now provide an abstract framework upon which relational complexes can be defined. Note that `IntMap` and `IntMatrix` are abstract datatypes which may also be implemented by relational database access using JDBC.

4.4.2. *Relational Complexes in Java.* A relational complex in Java is simply an interface `Complex<T>`, with at least the methods

```
Set<T> getCells()
```

and

```
IntMatrix<T,T> getBoundary().
```

The boundary returned must yield a partial zero matrix if multiplied to itself, hence the following must not fail if `bd=getBoundary(someComplex)`:

```
for(Integer i : bd.mul(bd).values()){
   if(i.intValue()!=0)
      throw new Exception("not a complex");
}
```

Note that explicit mappings to `null` are not allowed, hence a `NullPointerException` at `i.intValue()` cannot occur.

Other useful methods for a complex are `getCells(int dim)` to return the set of cells of some dimension `dim` as a set and, conversely, a method `dimension(T cell)` returning the dimension of a cell.

As such complexes are meant as a data model for computer aided engineering, the modifiers are very important methods. First of all not all complexes may be editable, indicated by a method `isEditable()`. The elementary optional modifier methods are then

```
addCell(T cell, Map<T,Integer> boundary)
```

and

```
removeCell(T cell)).
```

which permit the modification of editable complexes and throw an `UnsupportedOperationException` (hence are optional) otherwise. The method `addCell` throws an `IllegalArgumentException` if the boundary of the specified boundary is not a partial zero map and `removeCell`, on the other hand, throws an `IllegalStateException` if the specified cell is in the boundary of some other cell. We call these methods the elementary *Euler-Poincaré operators*. They maintain the fundamental complex property, thus prevent the construction of anything which is not a complex and always permit the iterative construction of a complex.

Having indicated how relational complexes can be implemented, we will return to theory in the following sections. Note that `cmpl.getBoundary().keySet()` for a complex `cmpl` returns a `Set<Pair<T,T>>`, in other words, a binary relation on the cells of `cmpl`. The topological meaning of this relation will become evident in what follows.

5. Categories of topological spaces

As we are interested not only in modeling spatial information, but also desire to take care of relations between spaces, we adopt the point of view of categories and functors. Hence, these "relations" will be modeled by morphisms in a category.

5.1. **Relational databases as topological spaces.** In this section, we introduce a category of a special kind of relational database schemes and a category of topological spaces and show that there is an equivalence of categories between them. This merely simply reproduces results of [3] in categorial terms with a slight generalization: Whereas Alexandrov assumes a transitive and reflexive relation on a set, here any relation on a set is accepted, because its transitive and reflexive closure can always be computed. As we want to store the relation which generates a topology into a relational database, we do not want to store redundant pairs (a, c) which can be computed from already stored pairs (a, b) and (b, c).

Definition 5.1 (**DTop**)**.** *Let* **DTop** *be the following category: The objects of* **DTop** *are pairs (X, R), where X and R are sets such that R is a binary relation on X, i.e. $R \subseteq X^2$. A morphism*

$$f\colon (X, R) \to (Y, S)$$

is a map $f\colon X \to Y$ between the underlying sets such that

$$(f \times f)(R) \subseteq S^*,$$

where $f \times f\colon X \times X \to Y \times Y$ is the map

$$(u, v) \mapsto (f(u), f(v)),$$

and S^* is the reflexive and transitive closure of S. We call the objects of **DTop** topological datatypes*, and the morphisms are called* continuous database maps*. We then call a family $\{(X_i, R_i)\}_{i\in I}$ of topological datatypes a* topological database*.* □

Remark 5.2. A topological datatype (X, R) can be viewed in a natural way as a topological database $\{(X, R)\}$. The way in which an object (X, R) of **DTop** is to be seen as an actual database is to have a table for X and a table for R which encodes R as a relation on X.

Remark 5.3. A topological datatype (X, R) can be considered as a simple graph. However, a morphism of simple graphs $f\colon (X, R) \to (Y, S)$ fulfills the stronger condition

$$(f \times f)(R) \subseteq S \subseteq S^*.$$

Definition 5.4 (**Alex**)**.** *Let* **Alex** *be the following category: The objects of* **Alex** *are the topological spaces $(X, \mathcal{T}_X)$ having the property that the intersection of an arbitrary set $\mathcal{U} \subseteq \mathcal{T}_X$ of open subsets of X is open, i.e.*

$$\bigcap_{U\in\mathcal{U}} U \in \mathcal{T}_X. \tag{4}$$

The morphisms of **Alex** *are the continuous maps. An object of* **Alex** *is called an* Alexandrov space. □

Alexandrov calls the topological spaces satisfying property (4) *discrete* spaces [3].

Example 5.5 (Finite topological spaces)**.** Any finite topological space is an Alexandrov space, as $\mathfrak{U}$ can only contain finitely many open sets, and the intersection of finitely many open sets is always open.

The next aim of this section is to prove

Theorem 5.6 (Attributed to Alexandrov)**.** *There is an equivalence of categories*

$$F\colon \mathbf{DTop} \to \mathbf{Alex}.$$

The proof of Theorem 5.6 is performed in several steps after some preparation. First, consider the map

$$\mathrm{St}_{R^*}\colon \mathfrak{P}(X) \to \mathfrak{P}(X),\ A \mapsto \{x \in X \mid \exists\, a \in A\colon aR^*x\},$$

where $\mathfrak{P}(X)$ denotes the power set of X.

Definition 5.7 (Star neighbourhood)**.** *Let (X, R) be an object of* **DTop** *and A a subset of X. The set $\mathrm{St}_{R^*}(A) \subseteq X$ is called the* star neighbourhood *of A. If $A = \{x\} \subseteq X$, then we will write $\mathrm{St}_{R^*}(x)$ instead of $\mathrm{St}_{R^*}(\{x\})$.* □

Proposition 5.8. *Let (X, R) be a topological datatype. Then for any $A \subseteq X$ hold true:*

$$A \subseteq \mathrm{St}_{R^*}(A) \tag{5}$$

$$\mathrm{St}_{R^*}(\mathrm{St}_{R^*}(A)) = \mathrm{St}_{R^*}(A). \tag{6}$$

Proof. (5) is clear, as the diagonal R^0 is a subset of R^*.

(6) The inclusion $\supseteq$ follows from (5).

For $\subseteq$, let $x \in \mathrm{St}_{R^*}(\mathrm{St}_{R^*}(A))$. Then there exists a $y \in \mathrm{St}_{R^*}(A)$ such that xR^*y. For this $y \in \mathrm{St}_{R^*}(A)$ there is now an element $z \in A$ such that yR^*z. Because R^* is transitive, it follows that xR^*z, thus $x \in \mathrm{St}_{R^*}(A)$. □

We now define the functor F.

On objects. For an object (X, R) let $F(X, R)$ be the pair $(X, \operatorname{im} \mathrm{St}_{R^*})$, where $\operatorname{im} \mathrm{St}_{R^*}$ is the image of the map St_{R^*} considered above. This yields a well-defined map between objects:

Lemma 5.9. *$F(X, R)$ is an Alexandrov space.*

Proof. We check that $F(X, R)$ is a topological Alexandrov space by going through the axioms for $\operatorname{im} \mathrm{St}_{R^*}$.

(1) It is easily verified that $X = \mathrm{St}_{R^*}(X)$ and $\emptyset = \mathrm{St}_{R^*}(\emptyset)$.

(2) Let $\mathcal{A} \subseteq \mathfrak{P}(X)$. The following identities hold true:

$$\mathrm{St}_{R^*}\left(\bigcup_{A \in \mathcal{A}} \mathrm{St}_{R^*}(A)\right) = \bigcup_{a \in \mathcal{A}} \mathrm{St}_{R^*}(A) \tag{7}$$

$$\mathrm{St}_{R^*}\left(\bigcap_{A \in \mathcal{A}} \mathrm{St}_{R^*}(A)\right) = \bigcap_{a \in \mathcal{A}} \mathrm{St}_{R^*}(A) \tag{8}$$

The first equality is readily checked. For the second, the inclusion $\supseteq$ follows immediately from the expression (5). For the inclusion $\subseteq$, let x be an element of the left-hand side. Then $x\,R^*\,y$ for some $y \in \mathrm{St}_{R^*}(A)$ for all $A \in \mathcal{A}$. But then $x \in \mathrm{St}_{R^*}(\mathrm{St}_{R^*}(A)) = \mathrm{St}_{R^*}(A)$, where the latter equality is given by equation (6). This shows that $F(X,R)$ is a topological space having the property of being an Alexandrov space. □

On morphisms. Let now $f\colon (X,R) \to (Y,S)$ be a continuous database map in **DTop**. We define $F(f)$ to be the underlying map of sets $X \to Y$.

Lemma 5.10. *The induced map*

$$F(f)\colon (X, \operatorname{im}\mathrm{St}_{R^*}) \to (Y, \operatorname{im}\mathrm{St}_{S^*})$$

between Alexandrov spaces is continuous.

Note that the map $F(f)\colon X \to Y$ is nothing but the map $f\colon X \to Y$ in the definition of a morphism in **DTop**. Therefore we will denote the map $F(f)$ between the topological spaces of the Lemma simply by f.

Proof. Let $V \in \operatorname{im}\mathrm{St}_{S^*}$, i.e. $V = \mathrm{St}_{S^*}(B)$ for some subset B of Y. We must show that $f^{-1}(V)$ is open in $(X, \operatorname{im}\mathrm{St}_{R^*})$. We claim that $f^{-1}(V) = \mathrm{St}_{R^*}(f^{-1}(V))$, in which case $f^{-1}(V)$ is indeed open.

By Proposition 5.8.(5) we have $f^{-1}(V) \subseteq \mathrm{St}_{R^*}(f^{-1}(V))$. In order to show the other inclusion, let $x \in \mathrm{St}_{R^*}(f^{-1}(V))$. Then there is a $u \in f^{-1}(V)$ such that xR^*u. This implies

$$f(x)\ T\ f(u),$$

where $T := (f \times f)(R^*)$. Let $T' := ((f \times f)(R))^*$. Then

$$T \subseteq T' \subseteq S^*. \tag{9}$$

The right inclusion holds because S^* is reflexive and transitive and as, by assumption, $(f \times f)(R) \subseteq S^*$. For the left inclusion let $x\ R^*\ y$ be such that $f(x)\ T\ f(y)$. Then there is a sequence $x = x_1, \ldots, x_n = y$ such that $x_i\ R\ x_{i+1}$ for all $i = 1, \ldots, n-1$. So the sequence $f(x) = f(x_0), \ldots, f(x_n) = f(y)$ has the property that

$$f(x_i)\ (f \times f)(R)\ f(x_{i+1})$$

for all $i = 1, \ldots, n-1$. This means that $f(x)\ T'\ f(y)$.

Thus, by (9), it follows that $f(x)\ S^*\ f(u)$. But then

$$f(x) \in \mathrm{St}_{S^*}(V) = \mathrm{St}_{S^*}(B) = V,$$

where the latter equality is due to Proposition 5.8.(6). Thus, $x \in f^{-1}(V)$ which proves the claim. □

Proof of Theorem 5.6. From Lemmata 5.9 and 5.10 we see that F is a well defined functor

$$\mathbf{DTop} \to \mathbf{Alex}.$$

Claim 1. *F is fully faithful.*

Let $\mathbf{X} = (X, R)$ and $\mathbf{Y} = (Y, S)$. Then

$$\mathrm{Hom}_{\mathbf{DTop}}(\mathbf{X}, \mathbf{Y}) \to \mathrm{Hom}_{\mathbf{Alex}}(F\mathbf{X}, F\mathbf{Y}), f \mapsto F(f), \tag{10}$$

is the map between the sets of morphisms induced by the functor F. We must show that it is bijective.

We do this by giving an inverse map. The map

$$G\colon \mathrm{Hom}_{\mathbf{Alex}}(F\mathbf{X}, F\mathbf{Y}) \to \mathrm{Hom}_{\mathbf{DTop}}(\mathbf{X}, \mathbf{Y}), f \mapsto f.$$

takes any continuous function

$$f\colon (X, \operatorname{im} \mathrm{St}_{R^*}) \to (Y, \operatorname{im} \mathrm{St}_{S^*})$$

to itself, considered as a map

$$f\colon (X, R) \to (Y, S).$$

This is indeed a morphism in **DTop**, i.e. $(f \times f)(R) \subseteq S^*$: Namely, let

$$(u, v) \in (f \times f)(R).$$

This means that there are elements $a, b \in X$ with $a\ R\ b$ and $u = f(a)$, $v = f(b)$. Now, the fact that

$$u \in U := \mathrm{St}_{S^*}(u)$$

implies that $a \in f^{-1}(U)$. But, $b \in \mathrm{St}_{R^*}(a)$, because $a\ R\ b$. Thus, also $b \in f^{-1}(U)$, since $\mathrm{St}_{R^*}(a) \subseteq f^{-1}(U)$. The latter holds true, because $f^{-1}(U)$ is open by continuity of f. Therefore $v \in U$, in other words: $u\ S^*\ v$. This proves that f is a morphism in **DTop**, and G is therefore well defined.

It is clear that $F(G(f)) = f$ resp. $G(F(f)) = f$ for $f \in \mathrm{Hom}_{\mathbf{Alex}}(F\mathbf{X}, F\mathbf{Y})$, resp. $f \in \mathrm{Hom}_{\mathbf{DTop}}(\mathbf{X}, \mathbf{Y})$. This implies that the map (10) is bijective.

Claim 2. *F is essentially surjective.*

For this, let $(X, \mathcal{T})$ be an Alexandrov space. We must show that $(X, \mathcal{T})$ is homeomorphic to a space of the form $(Y, \operatorname{im} \mathrm{St}_{R^*})$ for some relation R on Y.

Let $Y := X$, and define the relation R as:

$$x\ R\ y :\iff y \in U_x,$$

where U_x is the intersection of all open subsets of $(X, \mathcal{T})$ containing x.

Note that R is reflexive and transitive, which implies $R = R^*$.

We have $U_x \in \mathcal{T}$, because $(X, \mathcal{T})$ is an Alexandrov space. Also,

$$U_x = \mathrm{St}_R(x),$$

so the sets $\mathfrak{U} := \{U_x \mid x \in X\}$ and $\mathfrak{S} := \{\mathrm{St}_R(x) \mid x \in X\}$ generate the same topology on X. Because the topology $\mathcal{T}$ is generated by $\mathfrak{U}$, this implies $\mathcal{T} = \operatorname{im} \mathrm{St}_R$. So, $(X, \mathcal{T})$ is not only homeomorphic, but even equal to $(X, \operatorname{im} \mathrm{St}_{R^*})$.

That F is an equivalence of categories now follows from Claim 1 and Claim 2. □

The following definition is a straightforward generalisation of the special case of partial orderings in [3].

Definition 5.11 (Topology generated by relation)**.** *Let R be a relation on a set X. We say that the topology $\mathcal{T} = \operatorname{im} \mathrm{St}_{R^*}$ on X is* generated by *R, and call $\mathcal{T}$ the* Alexandrov topology *of (X, R).* □

Remark 5.12. Note that it is quite tempting to disbelieve the statement of Theorem 5.6, because any two relations R and S on a set X satisfying $R^* = S^*$ generate the same topology on X. However, in this case, the two objects (X, R) and (X, S) are isomorphic in **DTop**, as will be seen in Theorem 5.13.

A more detailed description of the isomorphism classes of objects in **DTop** and **Alex** is given in the following theorem.

Theorem 5.13. *Let $\{(X, R), (Y, S)\}$ be a topological database. Then:*

(1) *$(X, \operatorname{im} \mathrm{St}_{R^*})$ and $(Y, \operatorname{im} \mathrm{St}_{S^*})$ are homeomorphic, if and only if there is a bijection $\phi\colon R^* \to S^*$, $(x, y) \mapsto (\phi_1(x, y), \phi_2(x, y))$ which takes stars to stars, i.e.*

$$y \in \mathrm{St}_{R^*}(x) \implies \phi_2(x, y) \in \mathrm{St}_{S^*}(\phi_1(x, y)).$$

(2) *Let $X = Y$. Then R and S generate the same topology, if and only if the bijection from* (i) *satisfies $\phi_1(x, y) = x$.*
(3) *For $X = Y$ holds true: $R^* = S^*$ if and only if $\mathrm{St}_{R^*} = \mathrm{St}_{S^*}$.*

Proof. Let for any relation T on a set M

$$\mathfrak{U}_T(M) := \{\mathrm{St}_{T^*}(x) \mid x \in M\}.$$

It is an easy fact that $\mathfrak{U}_T$ generates the topology St_{T^*}.

(1) Let $\phi\colon R^* \to S^*$ be a bijection taking stars to stars. Then for any $x \in X$

$$\Phi_x := \{\phi_2(x, y) \mid y \in \mathrm{St}_{R^*}(x)\} = \mathrm{St}_{S^*}(z),$$

where $z := \phi_1(x, y)$ for any $y \in \mathrm{St}_{R^*}(x)$ (note that this is independent of the choice of $y \in \mathrm{St}_{R^*}(x)$) by assumption). Indeed, we clearly have $\Phi_x \subseteq \mathrm{St}_{S^*}(x)$ by assumption, and if $w \in \mathrm{St}_{S^*}(z)$, then there is a (unique) $u \in X$ such that $x\ R^*\ u$, i.e. $w \in \Phi_x$. Then the map

$$X \to Y, \quad x \mapsto z \quad \text{(as above)}$$

induces a bijection

$$\Psi\colon \mathfrak{U}_R(X) \to \mathfrak{U}_S(Y),$$

hence a homeomorphism

$$(X, \operatorname{im} \mathrm{St}_{R^*}) \to (Y, \operatorname{im} \mathrm{St}_{S^*}).$$

Let, for the converse implication, $f\colon (X, \operatorname{im} \mathrm{St}_{R^*}) \to (Y, \operatorname{im} \mathrm{St}_{S^*})$ be a homeomorphism. Then f induces a map

$$\Psi\colon \mathfrak{U}_R(X) \to \mathfrak{U}_S(Y).$$

Namely, there is the map

$$\Psi\colon \mathfrak{U}_R(X) \to \operatorname{im} \mathrm{St}_{S^*},\ U \mapsto f(U).$$

The image of Ψ is $\mathfrak{U}_S(Y)$: $f(\mathrm{St}_{R^*}(x))$ is an open set containing $f(x)$, hence

$$f(\mathrm{St}_{R^*}(x)) \supseteq \mathrm{St}_{S^*}(f(x)).$$

And $f^{-1}(\mathrm{St}_{S^*}(f(x))$ is an open set containing x, hence

$$f^{-1}(\mathrm{St}_{S^*}(f(x))) \supseteq \mathrm{St}_{R^*}(x).$$

FIGURE 2

This implies

$$\begin{aligned}\mathrm{St}_{R^*}(x) &\subseteq f^{-1}(\mathrm{St}_{S^*}(f(x)))\\ &\subseteq f^{-1}(f(\mathrm{St}_{R^*}(x))) = \mathrm{St}_{R^*}(x),\end{aligned}$$

hence $\mathrm{St}_{R^*}(x) = f^{-1}(\mathrm{St}_{S^*}(f(x)))$, and we obtain the induced map $\Psi\colon \mathfrak{U}_R(X) \to \mathfrak{U}_S(Y)$.

Now, the map

$$R^* \to S^*,\ (x,y) \mapsto (f(x), f(y))$$

is clearly a bijection satisfying

$$y \in \mathrm{St}_{R^*}(x) \Rightarrow f(y) \in \mathrm{St}_{S^*}(f(x)),$$

i.e. takes stars to stars.

(2) By what we have seen in the proof of (1), we have

$$\mathfrak{U}_R(X) = \mathfrak{U}_S(X) \iff \forall x \in X \quad \phi_1(x,y) = x.$$

In this case, the topologies on X generated by R and S coincide.

(3) By the above, it holds true that the induced map $\Psi\colon \mathfrak{U}_R(X) \to \mathfrak{U}_S(Y)$ is the identity, if and only if for all $x \in X$ the stars $\mathrm{St}_{R^*}(x)$ and $\mathrm{St}_{S^*}(x)$ coincide, i.e. if and only if $\mathrm{St}_{R^*} = \mathrm{St}_{S^*}$. □

The following example shows that not every bijection $R^* \to S^*$ yields homeomorphic topological spaces.

Example 5.14 (Bijection between relations)**.** Consider the two graphs of Figure 2 which are transitive. There is obviously a bijection between the reflexive and transitive closures of the corresponding relations, as the numbers of oriented edges coincide. However, the two corresponding topological spaces are not homeomorphic, as the numbers of connected components differ.

Another useful characterisation of **DTop** is the following

Theorem 5.15. *The category* **DTop** *is equivalent to the category $\mathcal{G}_{rts}$ of reflexive and transitive simple graphs.*

The morphisms of the category $\mathcal{G}_{rts}$ are meant to be graph morphisms. Recall that for a simple graph $G = (V, E, d\colon E \to V \times V)$ the set of edges E consists of pairs of vertices: $E \subseteq V \times V$.

Proof. Let $\Gamma\colon$ **DTop** $\to \mathcal{G}_{rts}$ be the functor which maps (X, R) to the graph $\Gamma(X, R)$ whose set of vertices is X and set of edges is R^* (for pairs $(x, y) \in R^*$ let x be the origin and y the terminal vertex); and which maps topological database maps

$$f\colon (X, R) \to (Y, S)$$

to the natural graph morphism

$$\Gamma(f)\colon \Gamma(X, R) \to \Gamma(Y, S)$$

obtained by the condition

$$(f \times f)(R) \subseteq S^*.$$

We check that Γ is an equivalence. First, the induced map

$$\Gamma\colon \operatorname{Hom}((X,R),(Y,S)) \to \operatorname{Hom}(\Gamma(X,R),\Gamma(Y,S))$$

is bijective: because by a graph morphism

$$\phi\colon \Gamma(X,R) \to \Gamma(Y,S)$$

edges are taken to edges, in particular edges in R are mapped into S^*. Therefore, restricting to R induces a morphism in **DTop**

$$f\colon (X,R) \to (Y,S)$$

such that $\Gamma(f) = \phi$. This determines a map

$$\Delta\colon \operatorname{Hom}(\Gamma(X,R),\Gamma(Y,S)) \to \operatorname{Hom}((X,R),(Y,S))$$

which is readily seen to be the inverse map to Γ on morphisms. This means that Γ is fully faithful.

Let now $G = (V,E,d)$ be a reflexive, transitive, simple graph. Then (V,E) is obviously an object in **DTop** such that $\Gamma(V,E) = G$. So, Γ is essentially surjective. □

Remark 5.16. The fact that **DTop** is equivalent to a category of topological spaces means firstly that it is not only possible to create databases of the types of topological spaces considered without any loss of topological information, but also continuous maps $f\colon X \to Y$ can be stored in a lossless way, simply by taking the graph of f.

5.2. **Relational complexes in DTop.** Given a relational complex $\mathfrak{C}$, one obtains a topological datatype simply by taking the basis B and the relation $R_\partial \subseteq B \times B$ defined as

$$(a,b) \in R_\partial \iff \partial(a,b) \neq \uparrow, \tag{11}$$

where ∂ is the relational boundary of $\mathfrak{C}$. We sometimes will write $a \leq b$ instead of $(a,b) \in R_\partial$, because the relation defines a partial ordering on B.

Proposition 5.17. *The Alexandrov topology on the underlying set $\mathfrak{X}$ of a combinatorial cw-complex $\mathbb{X}$ induced by the relation R_∂ from (11) coincides with the combinatorial topology on $\mathbb{X}$.*

Proof. Let $(X,\sim)$ be a cw-complex for which $\mathbb{X} = X/\sim$, and $x \in \mathfrak{X}_n$ an n-cell of $\mathbb{X}$. By definition of the weak topology on X, any open neighbourhood of x in $\mathbb{X}$ for the combinatorial topology consists of cells from X_m with $m > n$. Any minimal open neighbourhood U_x of x consists of cells y whose boundary contain x. The latter means that $\partial(x,y) \neq \uparrow$. In other words, $U_x \subseteq \operatorname{St}_{R_\partial^*}(x)$.

Now, the set $\operatorname{St}_{R_\partial^*}(x)$ consists precisely of the cells of $\mathbb{X}$ at whose boundary lies x. This is clearly a minimal open neighbourhood of x in the combinatorial topology. Hence, $\operatorname{St}_{R_\partial^*}(x)$ is the smallest open neighbourhood of x, and the combinatorial topology is an Alexandrov topology, and moreover coincides with the topology generated by R_∂. □

If $M\colon \mathfrak{C} \to \mathfrak{C}'$ is a morphism of relational complexes, then the partial matrix $m :\subseteq B \times B' \to \mathbb{Z}$ does not, in general, allow in a natural way to define a corresponding map $B \to B'$. However, for any morphism of finite combinatorial cw-complexes $f\colon \mathbb{X} \to \mathbb{X}'$, there is the underlying map $\sigma_f\colon \mathcal{X} \to \mathcal{X}'$.

Proposition 5.18. *The map σ_f is continuous for the Alexandrov topologies on $(\mathcal{X}, R)$ and $(\mathcal{X}', R')$. This defines a functor*

$$F\colon \mathbf{CombCW} \to \mathbf{DTop}.$$

Proof. This is a direct consequence of Proposition 5.17 and the fact that σ_f is continuous for the combinatorial topologies. The functoriality is clear. □

As to the functorial relation between cw-complexes and **DChainComp**, note that a continuous map which is not cellular does not induce a map between the corresponding marked chain complexes. E.g. let the following map be the identity map on the unit interval:

Here, the boundary of any of the two edges on the left does not map to the boundary of its image, as would be required for any induced chain map.

If, on the other hand, a map $\phi\colon X \to X'$ of cw-complexes is cellular, it induces a well-defined chain map $\phi_*\colon \mathcal{C}(X) \to \mathcal{C}(X')$. The matrix entries of ϕ_* are "determined by topology", and the relation $\partial'\phi_* = \phi_*\partial$ follows naturally. However, this is not the case for combinatorial complexes. Consider, for example, the inversion map of an edge:

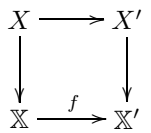
In **CW**, the change in orientation is encoded in the map between the points on the edges. The corresponding encoding in **DChainComp** is straightforward. But in **CombCW**, the edge object on the left maps to the edge object on the right, disregarding any informtion on orientation. Consequently, it is always possible to obtain a trivial encoding in **DChainComp** via the partial zero matrix. This morphism coincides on the sets of cells with the corresponding map in **DTop**.

If a relation $R \subseteq N \times M$ is the graph of a function $f\colon N \to M$, then we call the transposed relation $\{(m, n) \in M \times N \mid m = f(n)\}$ the *inverted graph* of f.

Proposition 5.19. *Let $f\colon \mathbb{X} \to \mathbb{X}'$ be a cellular map between combinatorial cw-complexes. Then there is a morphism $M_f\colon \mathfrak{C}(\mathbb{X}) \to \mathfrak{C}(\mathbb{X}')$ of relational complexes whose underlying partial matrix $m_f :\subseteq B' \times B$ is defined precisely on the inverted graph of f and whose entry in $x = (f(b), b) \in B'_j \times B_i$ satisfies*

$$m_f(x) \neq 0 \Leftrightarrow i = j.$$

Proof. The combinatorial cw-complexes $\mathbb{X}'$ and $\mathbb{X}$ are quotients of cw-complexes X and X'. Because the quotient maps are open, the map f lifts to a cellular map between cw-complexes such that the diagram commutes:



The morphism of relational complexes $\mathfrak{C}(X) \to \mathfrak{C}(X')$ has the desired property. As $\mathfrak{C}(X) = \mathfrak{C}(\mathbb{X})$ and $\mathfrak{C}(X') = \mathfrak{C}(\mathbb{X}')$, the assertion follows. □

Remark 5.20. *From a computational point of view, the proof of Proposition 5.19 is not very satisfactory, because the entries of the partial matrix are obtained via the map between cw-complexes. Hence, the map between the relational complexes depends heavily on the choice of a map between cw-complexes.*

However, in order to have a control of whether or not a partial matrix $m_f\colon B' \times B \to \mathbb{Z}$ *defines a morphism of relational chain complexes* $\mathfrak{C}(\mathbb{X}) \to \mathfrak{C}(\mathbb{X}')$*, one can proceed in a more constructive manner.*

To this end, we may assume that the domain of definition of m_f *is the inverted graph of* f*. Let for* $b \in B$ $\mu_b^f := m_f(f(b), b)$*.*

The condition $\partial' \cdot M_f = M_f \cdot \partial$ *translates for* $(b', b) \in B'_j \times B_i$ *to*

$$(\partial' \cdot m_f)(b', b) = (m_f \cdot \partial)(b', b) \tag{12}$$

The left side of (12) equals

$$\sum_{\ell \in B'} \partial'(b', \ell) \cdot m_f(\ell, b) = \partial'(b', f(b)) \cdot m_f(f(b), b)$$
$$= \begin{cases} [f(b) : b'] \cdot \mu_b^f, & j = i - 1 \wedge b' \leq f(b) \\ 0, & j < i - 1 \wedge b' \leq f(b) \\ \uparrow, & \text{otherwise.} \end{cases}$$

The right hand side of (12) is clearly

$$R(b', b) := \sum_{\ell \in B_j \cap f^{-1}(b')} m_f(b', \ell) \cdot \partial(\ell, b)$$

which is defined if and only if $j \leq i - 1$ *and* $b' \leq f(b)$*. In the case* $j < i - 1$*, it holds true that* $R(b', b) = 0$*, whereas for* $j = i - 1$ *we have*

$$R(b', b) = \sum_{\ell \in B_{i-1} : f(\ell) = b'} [b : \ell] \cdot \mu_\ell^f,$$

if $b' \leq f(b)$*. Hence, (12) translates for* $(b', b) \in B'_{i-1} \times B_i$ *with* $b' \leq f(b)$ *into the condition:*

$$[f(b) : b'] \cdot \mu_b^f = \sum_{\ell \in B_{i-1} : f(\ell) = b'} [b : \ell] \cdot \mu_\ell^f \tag{13}$$

which is a linear equation with integer coefficients in the unknowns μ_ℓ^f *and* μ_b^f*. It clearly has non-trivial solutions. Note that only those* μ_ℓ^f *occur in (13) for which* $(f(\ell), \ell) \in B'_k \times B_k$*. It is now easy to see that for fixed* $(b', b) \in B'_0 \times B_1$ *(i.e.* $i = 1$*) there are infinitely many solutions. These are inserted into the right hand sides of the equations for* $i = 2$ *in order then to obtain recursively* $(i \to i + 1)$ *all* μ_b^f *with* $b \in B$*.*

Lemma 5.21. *Assume that on the right hand side in (13) all* $\mu_\ell \neq 0$*. If the right hand side vanishes, then so does* $[f(b) : b']$*.*

Proof. This follows from the fact that f is induced by a cellular map of cw-complexes ϕ such that the induced chain map ϕ_* has the property $\phi_*(\ell) = \mu_\ell^f \cdot \phi(\ell)$, and the like for $b \in B_i$. In particular, this means that $\mu_b^f \neq 0$. Hence, $[f(b) : b']$ vanishes, if the right hand side of (13) does. □

Remark 5.22. *Lemma 5.21 implies a freedom of choice for those μ_b^f for which the right hand side of (13) vanishes, provided that $(f(b), b) \in B'_i \times B_i$.*

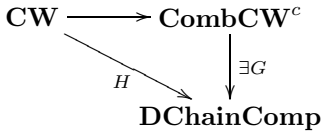
In order to obtain a functor, it is necessary to have $M_{g\circ f} = M_g \cdot M_f$ for morphisms $f\colon \mathbb{X} \to \mathbb{X}'$ and $g\colon \mathbb{X}' \to \mathbb{X}''$. This translates into the condition

$$\mu_b^{g\circ f} = \mu_{f(b)}^g \cdot \mu_b^f \tag{14}$$

for $b \in B$.

The subcategory of **CombCW** consisting of combinatorial complexes with cellular maps will be denoted by $\mathbf{CombCW}^c$.

Theorem 5.23. *There is a factorisation of the functor* $\mathbf{CW} \to \mathbf{DChainComp}$ *via* $\mathbf{CombCW}^c$*:*



where H is the straightforward functor to relational chain complexes.

Proof. It suffices to prove condition (14) under the rule that $\mu_b^f = 1$ whenever there is a freedom of choice as stated in Remark 5.22. In particular, $b \in B_0$ implies $\mu_b^f = 1$, and (14) holds true in this case. We proceed by induction: assume that (14) holds true for $i \in \mathbb{N}$, and let $b \in B_{i+1}$, $b'' \in B''_i$, $b'' \leq g(f(b))$. Then

$$\begin{aligned}
[g(f(b)) : b''] \cdot \mu_b^{g\circ f} &= \sum_{k \in B_i : g(f(k)) = b''} [b : k] \cdot \mu_k^{g\circ f} \\
&\overset{i.h.}{=} \sum_{k \in B_i : g(f(k)) = b''} [b : k] \cdot \mu_{f}(k)^g \cdot \mu_k^f \\
&= \sum_{\ell \in B'_i : g(\ell) = b''} \sum_{k \in B_i : f(k) = \ell} [b : k] \cdot \mu_k^f \cdot \mu_\ell^g \\
&= \sum_{\ell \in B'_i : g(\ell) = b''} [f(b) : \ell] \cdot \mu_b^f \cdot \mu_\ell^g \\
&= [g(f(b)) : b''] \cdot \mu_{f(b)}^g \cdot \mu_b^f
\end{aligned}$$

where the second equality holds true by the induction hypothesis. If $[g(f(b)) : b''] \neq 0$, then (14) follows. Assume $[g(f(b)) : b''] = 0$. Then $\mu_b^{g\circ f} = \mu_{f(b)}^g = 1$ by the two corresponding instances of freedom of choice. But also for $\ell \in g^{-1}(b'')$,

$$\begin{aligned}
[f(b) : \ell] \cdot \mu_b^f &= \sum_{k \in B_i : f(k) = \ell} [b : k] \cdot \mu_k^f \\
&= \sum_{k \in B_i : g(f(k)) = b''} [b : k] \cdot \mu_k^f \\
&= [g(f(b)) : b''] \cdot \mu_b^{g\circ f} = 0.
\end{aligned}$$

Hence, again by freedom of choice implies $\mu_b^f = 1$. □

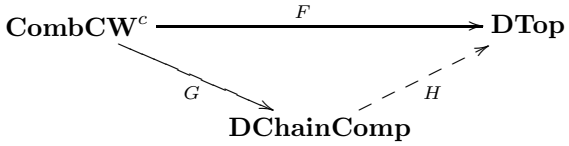
Remark 5.24. *There is no natural way of defining a functor* $\mathbf{DChainComp} \to \mathbf{DTop}$*, because the partial matrices representing morphisms in* **DChainComp** *do not naturally yield maps between topological datatypes. Namely, given a morphisms*

$M\colon \mathfrak{C} \to \mathfrak{C}'$ of relational complexes, then the corresponding partial matrix $m\colon \subseteq B \times B' \to \mathbb{Z}$ yields a relation $R_m \subseteq B \times B$, where

$$(b, b') \in R_m \Leftrightarrow m(b, b') \neq \uparrow .$$

If this relation is the graph of a function $B \to B'$, then it is possible to obtain a continuous database map.

Corollary 5.25. *The functor $F\colon$* **CombCW** $\to$ **DTop** *is essentially surjective onto the subcategory of* **DTop** *whose objects consist of simple, transitive, directed acyclic graphs. Its restriction to* $\mathbf{CombCW}^c$ *factorises as follows:*



where G is as in Theorem 5.23, and H is a functor from the subcategory of **DChainComp** *whose objects are the relational complexes and the morphisms are represented by partial matrices m such that R_m, as defined in Remark 5.24, is the graph of a function.*

Proof. F is a functor by Proposition 5.18. It is clear that the objects in the image of F are simple, transitive and directed acyclic graphs. Now, let $f\colon B \to B'$ be a morphism in **DTop** between the topological spaces $\mathcal{X}$ and $\mathcal{X}'$ associated to the combinatorial cw-complexes $\mathbb{X}$ and $\mathbb{X}'$. We need to show that there is a morphism $\phi\colon \mathbb{X} \to \mathbb{X}'$ such that $F(\phi) = f$. The map $\phi := f$ does the job, because the morphisms in **CombCW** are precisely the continuous maps.

It is clear that G takes morphisms $f\colon \mathbb{X} \to \mathbb{X}'$ in $\mathbf{CombCW}^c$ to morphisms of relational complexes whose corresponding partial matrices m are such that R_m is the graph of a function. In this case, the function $r_m\colon \mathcal{X} \to \mathcal{X}'$ is a morphism

$$r_m\colon (\mathcal{X}, R_\partial) \to (\mathcal{X}', R_{\partial'})$$

in **DTop**. This follows from the fact, that r_m coincides with the map f, and the latter is continuous for the corresponding Alexandrov topologies by Proposition 5.17. This constructs the functor $H \circ G$.

In order to show that $H \circ G = F$, observe first that by Proposition 5.17, the topological spaces associated by $H \circ G$ and F to a given combinatorial cw-complex coincide, whence the equality on objects. As to equality on morphisms, we have already observed that $r_m = f$, using the notation above. □

5.3. **Complexity of DTop.** The definition of **DTop** is straightforward from the observation, that every Alexandrov space can be characterized by a relation. It simply stores such a relation, which gives a storage complexity of $\mathcal{O}(n^2)$, where n is the cardinality of the underlying point set. On the first glance, one might consider this very inefficient for practical purposes and suppose that more subtle techniques exist to save storage space. No such technique exists, however, to improve this asymptotical behaviour. It has been shown that $\mathcal{O}(n^2)$ is the best possible storage complexity upper bound for every data structure for storing arbitrary topologies [5]. In *practical* cases, however, the topologies are less complex, so the problem of quadratic complexity will not arise there. Topologies derived from architectural

spaces, for example, mostly have linear storage complexity $\mathcal{O}(n)$ in **DTop** and can therefore be efficiently stored in **DTop** as well as in **DChainComp** [5].

6. Topological constructions

The motivation for this section is to define a query language for topological databases. This can be achieved by transforming the standard relational database queries into constructions of topological spaces out of given ones.

6.1. **The lattice of topologies.** Topologies on a given set have the handy property of being partially ordered by inclusion. Therefore topologies on a set can be compared and one considered best for some purpose can be chosen.

Definition 6.1 (Ordering of Topologies)**.** *Let $\mathcal{S}$ and $\mathcal{T}$ be topologies on a set X. Then $\mathcal{S}$ is said to be* coarser *than $\mathcal{T}$, if $\mathcal{S} \subseteq \mathcal{T}$, and $\mathcal{T}$ is also called* finer *than $\mathcal{S}$.*

If $\mathcal{S}$ is coarser (finer) than $\mathcal{T}$, then the topological space $(X, \mathcal{S})$ is also said to be coarser (finer) than $(X, \mathcal{T})$. □

Note that $\mathcal{T}$ is finer than $\mathcal{S}$ if and only if the identical map on the underlying set X

$$\mathrm{id}_X \colon X \to X, \quad x \mapsto x,$$

defines a continuous map $\mathrm{id}_X \colon (X, \mathcal{T}) \to (X, \mathcal{S})$.

In the case of Alexandrov topologies, there is a criterion for deciding whether one topology is finer than another. For simplicity of notation, we define

$$\mathcal{T}_R := \operatorname{im} \mathrm{St}_{R^*}$$

to be the topology generated by the relation R.

Lemma 6.2. *Let R and S be relations on X. Then $\mathcal{T}_R$ is finer than $\mathcal{T}_S$, if and only if $R \subseteq S^*$. Furthermore, $\mathcal{T}_R$ equals $\mathcal{T}_S$, if and only if $R^* = S^*$.*

Proof. Note first that $(\mathrm{id}_X \times \mathrm{id}_X)(R) = R$. Thus, we have the equivalence of the following statements

(1) $\mathcal{T}_R$ is finer than $\mathcal{T}_S$.
(2) $\mathrm{id}_X \colon (X, \mathcal{T}_R) \to (X, \mathcal{T}_S)$ is continuous.
(3) $(\mathrm{id}_X \times \mathrm{id}_X)(R) = R \subseteq S^*$.

Here the equivalence of (ii) and (iii) follows from Theorem 5.6.

This proves the first part of the lemma. The second part is an immediate consequence of the first part. □

The trivial topology $\{\emptyset, X\}$ is the coarsest topology that exists for X and the set of all subsets of X, called discrete topology, is its finest toplogy. It is well known that for every set $\mathfrak{T}$ of topologies for a set X there is an infimum $\inf \mathfrak{T} = \bigcap_{\mathcal{T} \in \mathfrak{T}} \mathcal{T}$, and a supremum $\sup \mathfrak{T}$, the topology generated by $\bigcup_{\mathcal{T} \in \mathfrak{T}} \mathcal{T}$.

Lemma 6.3. *Let $\mathfrak{T}$ be a family of Alexandrov topologies on a set X. Then $\inf \mathfrak{T}$ is an Alexandrov topology. If $\mathfrak{T}$ is finite, then also $\sup \mathfrak{T}$ is Alexandrov.*

Proof. The proof is straightforward. □

Lemma 6.3 states that there must exist relations generating the infimum and supremum topologies without making those relations explicit. The next theorem provides a construction of these relations.

Theorem 6.4 (with A. Höschele)**.** *Let $\{R_i\}_{i\in I}$ be a family of relations on a set X. Then $\bigcup_{i\in I} R_i$ generates $\inf\{\mathcal{T}_{R_i} \mid i \in I\}$. If I is finite, then $\bigcap_{i\in I} R_i^+$ generates $\sup\{\mathcal{T}_{R_i} \mid i \in I\}$, where R^+ denotes the transitive closure of a relation R on X.*

Proof. Let $S := \bigcup_{i\in I} R_i$. By Lemma 6.3, $\mathcal{T}_{\inf} := \inf\{\mathcal{T}_{R_i} \mid i \in I\}$ is Alexandrov, hence generated by a relation $R_{\inf}$. As for all $i \in I$ we have $\mathcal{T}_{\inf} \subseteq \mathcal{T}_{R_i}$, it follows by Lemma 6.2 that $R_i \subseteq R_{\inf}^*$. Hence, again Lemma 6.2 implies that $\mathcal{T}_{\inf} = \mathcal{T}_{R_{\inf}} \subseteq \mathcal{T}_S$.

On the other hand, all R_i are contained in S. Lemma 6.2 then implies that all $\mathcal{T}_{R_i}$ are finer than $\mathcal{T}_S$. Hence, $\mathcal{T}_S \subseteq \mathcal{T}_{\inf}$. This proves the first assertion.

Assume now I finite. Let $R := \bigcup_{i\in I} R_i^*$, and $\mathcal{T}_{\sup} := \sup\{\mathcal{T}_{R_i} \mid i \in I\}$. We want to prove that $\mathcal{T}_{\sup} = \mathcal{T}_R$. First, we have for all $i \in I$ that $R \subseteq R_i$ which yields $\mathcal{T}_R \supseteq \mathcal{T}_{R_i}$ for all $i \in I$, by Lemma 6.2, hence $\mathcal{T}_R \supseteq \mathcal{T}_{\sup}$.

For the converse inclusion: $\mathcal{T}_{\sup} \supseteq \mathcal{T}_{R_i}$ for all $i \in I$ yields $R_{\sup} \subseteq R_i^*$ for all $i \in I$, where $R_{\sup}$ is a relation generating $\mathcal{T}_{\sup}$ (which exists by Lemma 6.3). Hence, $R_{\sup} \subseteq R$, and $\mathcal{T}_{\sup} \supseteq \mathcal{T}_R$. The second assertion now follows, because

$$R = \left(\bigcap_{i\in I} R_i^+\right)^* .$$

□

If $\mathfrak{T}$ is not finite, then $\mathcal{T}_{\sup}$ need not be Alexandrov, as the following example shows:

Example 6.5 (Non-Alexandrov supremum)**.** Let $X = [0, 1]$ be the unit interval and $\mathcal{T}_n$ the topology generated by the set

$$\left\{\{0\}, \left(0, \frac{1}{n}\right), \left\{\frac{1}{n}\right\}, \ldots, \left\{\frac{n-1}{n}\right\}, \left(\frac{n-1}{n}, 1\right), \{1\}\right\}$$

where (x, y) denotes the open interval $\{r \in \mathbb{R} \mid x < r < y\}$. Then $\sup\{\mathcal{T}_n \mid n \in \mathbb{N}\}$ can be shown to be the Euclidian topology on X, which is not Alexandrov.

6.2. **Some examples of topological constructions.**

6.2.1. *Subspace topology.* Let $A \subseteq X$ be a subset of a topological space X. Recall that A carries the subspace topology which is simply obtained by intersecting A with all open sets of X.

Lemma 6.6. *Let (X, R) be a topological datatype and $A \subseteq X$. Then $A^2 \cap R^+$ generates the subspace topology for A.*

Proof. Let $i_A \colon A \to X$ be the natural inclusion map. As $A^2 \cap R^+ = (i_A \times i_A)^{-1}(R^+)$, the assertion is a special case of Theorem 6.9. □

The following example illustrates that in general, the computation of the transitive closure R^+ is necessary to generate the subspace topology.

Example 6.7 (by A. Höschele)**.** Let (X, R) be the topological database whose Alexandrov space depicted is in Figure 3. Then the subspace topology on $A = \{a, b\}$ does not contain the open set $\{b\}$ which is, however, contained in the discrete topology on A, the one generated by $A^2 \cap R = \emptyset$.

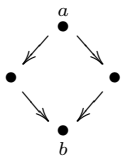


FIGURE 3

It will later be demonstrated that the transitive closure is necessary to compute the subspace topology, which is also the so-called initial topology of the inclusion mapping from A to X.

6.2.2. *Initial and final topologies.* As topological constructions are obtained by constructing initial or final topologies, we show how to construct these, first.

Definition 6.8 (Initial and final topology)**.** *Let $f_i\colon X_i \to Y$ $(i \in I)$ and $g_j\colon X \to Y_j$ $(j \in J)$ be maps between sets, where the $(X_i, \mathfrak{T}_{X_i})$ and $(Y_j, \mathfrak{T}_{Y_j})$ are topological spaces. Then the finest topology $\mathfrak{T}_f$ on Y such that all f_i are continuous is called the* final *topology of the family $\{f_i\}_{i\in I}$ on Y, and the coarsest topology $\mathfrak{T}_g$ on X such that all g_j are continuous is called the* initial *topology of the family $\{g_j\}_{j\in J}$ on X.* □

Theorem 6.9. *Let $f_i\colon X_i \to Y$ $(i \in I)$ and $g_j\colon X \to Y_j$ $(j \in J)$ be maps where the $\{(X_i, R_i)\}_{i\in I}$ and $\{(Y_j, S_j)\}_{j\in J}$ are topological databases. Then*

$$S_f := \bigcup_{i\in I} (f_i \times f_i)(R_i)$$

generates the final topology of $\{f_i\}_{i\in I}$ on Y. If J is finite, then

$$R_g := \bigcap_{j\in J} (g_j \times g_j)^{-1}(S_j^*)$$

generates the initial topology of $\{g_j\}_{j\in J}$ on X. If, additionally, all g_j are injective, then the initial topology of $\{g_j\}_{j\in J}$ on X is generated by

$$R_{g,+} := \bigcap_{j\in J} (g_j \times g_j)^{-1}(S_j^+).$$

If the family $\{g_j\}_{j\in J}$ consists of a single map g into the topological datatype (Y, S), and if that map is surjective, then the initial topology on X is generated by

$$R = (g \times g)^{-1}(S).$$

Proof. As it is seen that the final topology of the family $\{f_i\}$ is the infimum of the final topology of the final topologies of the single maps f_i, and the initial topology of the family $\{g_j\}$ is the supremum of the initial topologies of the single maps g_j, the proof is reduced to the case of single maps $f\colon X \to Y$ (with relation R on X) and $g\colon X \to Y$ (with relation S on Y).

In that case, we have

$$R_f = (f \times f)(R)$$
$$S_g = (g \times g)^{-1}(S^*)$$
$$S_{g,+} = (g \times g)^{-1}(S^+).$$

Final topology. The map $f\colon (X, R) \to (Y, R_f)$ is clearly continuous. Let now R_Y be any relation on Y such that $f\colon (X, R) \to (Y, R_Y)$ is continuous. Then, by definition, $R_f \subseteq R_Y^*$, hence $\mathcal{T}_{R_f}$ is finer than $\mathcal{T}_{R_Y}$. This means that $\mathcal{T}_{R_f}$ is the finest topology such that f is continuous, hence final.

Initial topology. The map $(X, S_g) \to (Y, S)$ is continuous, as $(g \times g)(S_g) \subseteq S^*$. Let now S_X be any relation on X such that $f\colon (X, S_X) \to (Y, S)$ is continuous. Then $(g \times g)(S_X) \subseteq S^*$ implies that $S_X \subseteq (g \times g)^{-1}(S^*) = S_g$, hence $\mathcal{T}_{S_g}$ is coarser than $\mathcal{T}_{S_X}$. Thus, $\mathcal{T}_{S_g}$ is the coarsest topology on X such that g is continuous.

g **injective.** We must show $S_{g,+}^* = S_g^*$. The inclusion $\subseteq$ is obvious. For the other inclusion, first observe that S_g is reflexive and transitive. Let $(a, b) \in S_g$. We may assume that $a \neq b$, as obviously $(a, a) \in S_{g,+}$. Now, $(g(a), g(b)) \in S^*$, and the injectivity of g implies $g(a) \neq g(b)$, i.e. $(g(a), g(b)) \in S^+$. Hence, $(a, b) \in S_{g,+}$. As one checks that $S_g = S_g^*$, the inclusion $\supseteq$ is proven.

g **surjective.** We know that $R' := (g \times g)^{-1}(S^*)$ is reflexive, transitive and generates the initial topology. Clearly, $R^* \subseteq R'$. We now show that $R' \subseteq R^*$. Let $(a, b) \in R'$. Then $(g(a), g(b)) \in S^n$ for some $n \in \mathbb{N}$. This means that there is a sequence

$$y_0 = g(a), y_1, \ldots, y_n = g(b)$$

such that $(y_i, y_{i+1}) \in S$ for all $i = 0, \ldots, n-1$. As g is surjective, this means that there exist $a_0 = a, a_1, \ldots, a_n = b$ such that $g(a_i) = y_i$ and $(a_i, a_{i+1}) \in R$ for all $i = 0, \ldots, n-1$. Thus, $(a, b) \in R^*$. □

The case of a surjective map g in Theorem 6.9 does not generalise to a family of more than one surjective maps $\{g_j\}_{j \in J}$.

Example 6.10 (Initial map requiring transitive closure)**.** Let $X = Y = \{a, b, c, d\}$, and $S_1 = \{(a, b), (b, d)\}$, $S_2 = \{(a, c), (c, d)\}$ relations on Y. The initial topology on X for the family consisting of the two identical maps

$$g_1 = g_2\colon X \to X, \quad x \mapsto x,$$

is generated by the relation $S_1^+ \cap S_2^+ = \{(a, d)\}$, whereas $S_1 \cap S_2 = \emptyset$ generates a different topology on X.

6.2.3. *Overwiew on relational algebra.* The query language we have in mind is the relational algebra. According to [9, p.63] only the semantics of the relational operators matter — their notation, however, is arbitrary. We first introduce the notion of a 'relational database' and other associated notions.

Definition 6.11 (Relational Schema)**.** *Let $\mathcal{D}$ be a fixed set of sets, which we call the* domains *of a relational database system. Each set $D \in \mathcal{D}$ is called an* elementary data type*. We then call a mapping $\mathcal{R} : \mathcal{A} \to \mathcal{D}$ from a (usually finite) set $\mathcal{A}$ of attribute names to the domains a* relational schema *for $\mathcal{D}$. We will often denote a relational schema $\mathcal{R} : \{a_1, \ldots, a_n\} \to \mathcal{D}$ by $\mathcal{R}[a_1 : D_1, \ldots, a_n : D_n]$, where $\mathcal{R}(a_i) = D_i$ holds for all $i = 1, \ldots, n$.* □

Note that this definition differs somewhat from standard literature, where the set $\mathcal{A}$ of attribute symbols is called 'relational schema'. But then an additional mapping $\mathrm{dom}_{\mathcal{A}} : \mathcal{A} \to \mathcal{D}$ is assumed which maps each attribute symbol a to its domain $\mathrm{dom}_{\mathcal{A}}(a)$ [8]. In this case, however, the complete specification of a relational schema would be the pair $(\mathcal{A}, \mathrm{dom}_{\mathcal{A}})$. But as the attribute set $\mathcal{A}$ is already given by the mapping, we see no need to explicitely mention it again and, hence, consider the mapping alone a relational schema.

Such relational schema is now used to define the data.

Definition 6.12 (Tuple, Relation, Relational Database)**.** *Let $\mathcal{R} : \mathcal{A} \to \mathcal{D}$ be a relational schema. Then a mapping $t : \mathcal{A} \to \bigcup \mathcal{D}$ is called a* tuple *to $\mathcal{R}$, if $t(a) \in \mathcal{R}(a)$ holds for each attribute symbol $a \in \mathcal{A}$. If G is a set of tuples to $\mathcal{R}$, then the pair $(\mathcal{R}, G)$ is called a* relation *to $\mathcal{R}$. $\mathcal{R}$ is often called the* header *and G is called the* body *or the* graph *of the relation (and is in general assumed finite). A* relational database *is simply a family $\{(\mathcal{R}_i, G_i) \mid i \in I\}$ of such relations over a finite index set I which consists of the table names.* □

To store, modify and retrieve information in relational databases a query language is needed. We will use relational algebra which provides us with the basic query operators.

Definition 6.13 (Relational algebra)**.** *We will call a query language $\mathfrak{L}$ for relational databases a* relational algebra *if $\mathfrak{L}$ it consists at least of the operators*

1. *constant domain values, tuples and relations,*
2. *theta-selection $\sigma_\Theta(R)$ of a relation R,*
3. *projection $\pi_{\mathcal{A}}(R)$ and renaming $\beta_{(a_i \leftarrow b_i)}(R)$ of a relation R,*
4. *cartesian product $R \times S$ or, alternatively, natural join $R \bowtie S$ of relations R and S and*
5. *union $R \cup S$, intersection $R \cap S$ and set difference $R \setminus S$ of relations R and S, if both are union compatible, hence have the same relational schema.*

$\mathfrak{L}$ is called a relational algebra with transitive closure *if for a relational schema for binary relations $\mathcal{R}[a : D,\, b : D]$ there is a query expression $C(R)$ which computes the transitive closure R^+ for every relation R to $\mathcal{R}$.* □

Note that the transitive closure expression only depends on the relational schema and not on the occurence of the given relation. It is well known, that with the operators alone, listed in the definition, it is not possible to formulate such a query for the transitive closure [8].

Except for the trivial constant queries, we will now show the topological versions of the above listed operations and, additionally, of the quotient relation $R/\mathcal{A}$—the set of the fibres of the projection $\pi_{\mathcal{A}}(R)$, called *framing a relation* by Codd and realized in SQL by `group by`-clauses.

6.2.4. *Selection in topological databases.* Our first example for a transformation of a relational database query into a topological construction will be the select-operator from relational algebra.

Definition 6.14 (topological select operator)**.** *Let (X, R) be a topological datatype and Θ a predicate expression to the relational schema of X. Then the query $\sigma_\Theta(X) := \{t \in X \mid t \text{ satisfies } \Theta\}$ is called the* theta-select *of X with Θ. Hence we call*

$$\underline{\sigma}_\Theta(X, R) := (\sigma_\Theta(X), R^+ \cap \delta_L(\sigma_\Theta(X)) \times \delta_R(\sigma_\Theta(X)))$$

the topological theta-select *of* (X, R) *with* Θ. □

Note that this is a special case of Lemma 6.6. As in practical databases, the relation R will only contain pairs of key values for key attributes in X, the $\sigma_\Theta(X)$ of the cartesian product expressions must be made union compatible with R by adequate renaming and projection queries. These queries are denoted by δ_L for the factor on the left hand side and δ_R on the right hand side of the cartesian product.

6.2.5. *Final topological database queries.* The corollaries in the remainder of this section all are consequences of Theorem 6.9.

Corollary 6.15. *Let* (X, R) *be a topological datatype and* $\sim$ *an equivalence relation on* X*. Then* $(\pi \times \pi)(R)$ *generates the quotient topology on* $X/\sim$*, where* $\pi\colon X \to X/\sim$ *is the canonical projection.*

Proof. This is a direct consequence of Theorem 6.9, as the quotient topology is the final topology on $X/\sim$ for the canonical projection π. □

Remark 6.16. If X is a relation in a relational schema $\mathcal{R}$ and $\mathcal{A}$ is a subset of the attribute set $\mathcal{X}$ of $\mathcal{R}$, then one can define an equivalence relation $\sim_\mathcal{A}$ on the set of all tuples of X by saying that tuples s and t are equivalent, if their projections onto $\mathcal{A}$ coincide. The resulting relation $X/\sim_\mathcal{A}$ is then a set of aggregated tuples of a corresponding relational schema $\mathcal{R}/\sim_\mathcal{A}$. In this way, the `group-by`-operator in SQL gets a topological meaning as a quotient map of topological datatypes.

The quotient topology is strongly related to the topology of the projection. We first recall that the projection $\pi_\mathcal{A}(X)$ of a relation X with attributes $\mathcal{X}$ is known as $\pi_\mathcal{A}(X) := \{\pi_\mathcal{A}(t) \mid t \in X\}$. Every record t in X is projected onto the attributes in $\mathcal{A}$. So we have a mapping $\pi_\mathcal{A} : X \to \pi_\mathcal{A}(X)$ from each record of the given relation X to a record of the resulting relation $\pi_\mathcal{A}(X)$, which therefore gets the final topology.

Definition 6.17 (Topological projection)**.** *Let* (X, R) *be a topological datatype with attributes* $\mathcal{X}$ *and* $\mathcal{A}$ *a subset of* $\mathcal{X}$*. We assume without loss of generality that* R *consists of pairs of records from* X*. Then, as* $(\pi_\mathcal{A} \times \pi_\mathcal{A})(R)$ *generates the topology on* $\pi_\mathcal{A}(X)$*, we call*

$$\underline{\pi}_\mathcal{A}(X, R) := (\pi_\mathcal{A}(X), (\pi_\mathcal{A} \times \pi_\mathcal{A})(R))$$

the topological projection *of* (X, R) *onto* $\mathcal{A}$. □

Note that the actual relational schema of R is not relevant for Definition 6.17 and the topological projection is unique up to homeomeorphism. If R does not consist of pairs of records from X, which is very likely to be the case in normalized relations, then we replace R by the expression $\pi_{1,2}((X \times X) \bowtie_\Theta R)$, where Θ is satisfied by every tuple $(x_a, x_b, (a, b))$ such that $(a, b) \in R$ and a references x_a as a foreign key and b is such reference to x_b. The projection $\pi_{1,2}$ turns $(x_a, x_b, (a, b))$ into (x_a, x_b).

To complete the list of standard unary relational operators we will briefly introduce renaming.

Definition 6.18 (topological renaming)**.** *Let* (X, R) *be a topological datatype,* f *a renaminig of attributes in* X *and* β_f *the according relational renaming operator for* X*. Then*

$$\underline{\beta}_f(X, R) := (\beta_f(X), (\beta_f \times \beta_f)(R))$$

is called the renaming *of a topological datatype.* □

The effect of renaming is obvious and only mentioned for completeness.

6.2.6. *Sums in topological databases.* Disjoint unions are known in SQL by the `union all` statements. These, however, are not proper disjoint unions but rather a lazy enumeration of the union of a set, avoiding the costly effort of removing duplicates. Codd calls such relations with duplicates *corrupted* [9]. In topology, however, there is a strict notion of disjoint unions, which we will now adopt to the relational model.

Corollary 6.19. *Let (X, R) and (Y, S) be union compatible topological datatypes with corresponding Alexandrov spaces $\mathbf{X}$ and $\mathbf{Y}$. Then the topology of $\mathbf{X} + \mathbf{Y}$ is generated by the relation $R \coprod S$ on $\{0\} \times X \cup \{1\} \times Y$, defined as*

$$\begin{aligned}(a,b) \in R \coprod S \\ \iff (a,b) = ((0,x),(0,y)) \text{ and } (x,y) \in R, \\ \text{or} \quad (a,b) = ((1,x),(1,y)) \text{ and } (x,y) \in S.\end{aligned}$$

Proof. This follows from Theorem 6.9, as the topology on $\mathbf{X}+\mathbf{Y}$ is the final topology of the natural inclusions $\mathbf{X} \to \mathbf{X} + \mathbf{Y}$ and $\mathbf{Y} \to \mathbf{X} + \mathbf{Y}$. □

In relational queries such disjoint unions must be made explicit by specifying an additional attribute `ref` to store the 0 and 1 values. We suggest a notation $X +_{\texttt{ref}} Y$ which says that X and Y are union compatible relations where an attribute named `ref` does not occur. Then we define two constant relations $\{\langle 0 : \texttt{ref}\rangle\}$ and $\{\langle 1 : \texttt{ref}\rangle\}$ and set

$$X +_{\texttt{ref}} Y := \{\langle 0 : \texttt{ref}\rangle\} \times X \cup \{\langle 1 : \texttt{ref}\rangle\} \times Y.$$

or, in SQL:

```
select 0 as ref, *
from X
union all
select 1 as ref, *
from Y;
```

6.2.7. *Glueing topological databases.* Another important construction is the glueing of topological spaces.

Corollary 6.20. *Let (X, R) and (Y, S) be union compatible topological datatypes with corresponding Alexandrov spaces $\mathbf{X}$ and $\mathbf{Y}$, and $\mathbf{A} \subseteq \mathbf{X}$ a topological subspace of $\mathbf{X}$. Let $f\colon \mathbf{A} \to \mathbf{Y}$ be a glueing map, hence $(f \times f)(R^+|_A) \subseteq S^*$, with $R^+|_A := R^+ \cap (A \times A)$. Then then the topology of the space $\mathbf{X} \coprod_f \mathbf{Y}$, obtained by glueing $\mathbf{Y}$ to $\mathbf{X}$ with glueing map f, is generated by*

$$(\pi \times \pi)(R \coprod S),$$

where $\pi\colon \mathbf{X} + \mathbf{Y} \to \mathbf{X} \coprod_f \mathbf{Y}$ is the canonical projection induced by the glueing with f.

Proof. This follows immediately from Corollaries 6.19 and 6.15. □

An SQL-expression corresponding to the glueing of topological spaces is straightforward, but leads to rather lengthy expressions which we will not particularise here. It is the query expression for the equivalence relation generated by f on $\mathbf{X} + \mathbf{Y}$ which is somewhat verbose.

6.2.8. *Topological product database.* Let $\{(X, R), (Y, S)\}$ be a topological database, and let $\mathbf{X}$ resp. $\mathbf{Y}$ be the corresponding Alexandrov spaces. Define the following relation $R \otimes S$ on $X \times Y$:

$$((a, b), (a', b')) \in R \otimes S \iff ((a, a'), (b, b')) \in R \times S.$$

Then, if we denote the diagonal relation Δ_X on X by R^0 (i.e. $a\, R^0\, b$, if and only if $(a, b) \in \Delta_X$, if and only if $a = b$), we obtain:

Corollary 6.21. *The product topology on* $\mathbf{X} \times \mathbf{Y}$ *is generated by*

$$(R \otimes S^0) \cup (R^0 \otimes S).$$

Proof. We will show that the topology generated by $T := (R \otimes S^0) \cup (R^0 \otimes S)$ is equivalent to the initial topology

$$T_\pi := (\pi_X \times \pi_X)^{-1}(R^*) \cap (\pi_Y \times \pi_Y)^{-1}(S^*)$$

for the projections $\pi_X \colon \mathbf{X} \times \mathbf{Y} \to \mathbf{X}$ and $\pi_Y \colon \mathbf{X} \times \mathbf{Y} \to \mathbf{Y}$.

1. $T^* \subseteq T_\pi^*$**.** Let $(a, b)T(a', b')$. We may assume that $(a, a') \in R$ and $b = b'$. As $(a, a') = (\pi_X \times \pi_X)((a, b), (a', b'))$ and $(b, b) \in S^*$, we have $(a, b)T_\pi(a', b')$.

2. $T^* \supseteq T_\pi^*$**.** Let $(a, b)T_\pi(a', b')$. Then $(a, a') = (\pi_X \times \pi_X)((a, b), (a', b')) \in R^*$, and also $(b, b') \in S^*$. Thus, $(a, b)R^* \otimes S^0(a', b)$ and $(a, b)R^0 \otimes S^*(a, b')$. By induction, it can be shown that

$$R^* \otimes S^0 = (R \otimes S^0)^* \quad \text{and} \quad R^0 \otimes S^* = (R^0 \otimes S)^*,$$

whence $(a, b)(R \otimes S^0 \cup R^0 \otimes S)^*(a', b')$, i.e. $(a, b)T^*(a', b')$. □

Together with the topological theta select it is then possible to compute topological inner joins. Outer joins, however, cannot be defined this way as the involved mappings are *partial* projections and partial mappings do not yield a unique initial topology. A further discussion of topological outer joins and a possible workaround can be found in [5].

Example 6.22 (Topological product in SQL)**.** Let X be a relation with relational schema $\{\underline{id}, \ldots\}$ and Y another relation with $\{\underline{nr}, \ldots\}$. Let R be a relation whith relational schema $\{\underline{ax} : X.id, \underline{bx} : X.id\}$ then (X, R) forms a topological database. A relation S having a relational schema $\{\underline{ay} : Y.nr, \underline{by} : Y.nr\}$ also gives a topological database (Y, S). The cartesian product $X \times Y$ is obviously given by the SQL-expression

```
create view XtimesY as
select * from X,Y;
```

Then, with the SQL-expression

```
create view RprodS as
select ax, bx, nr as ay, nr as by
from R,Y
union
select id as ax, id as bx, ay, by
from X,S;
```

the pair `(XtimesY, RprodS)` forms a topological database of the topological product, hence

$$(\texttt{XtimesY}, \texttt{RprodS}) \cong (X, R) \times (Y, S).$$

Remark 6.23. Note that according to Corollary 6.21 the product topology, despite it being an initial topology, can be computed without using the transitive closure operation.

6.2.9. *Union and intersection.* The union and the intersection of two sets both have characteristic inclusion mappings which define topologies on them. The union $A \cup B$ has the inclusions $i_A \colon A \hookrightarrow A \cup B$ and $i_B \colon B \hookrightarrow A \cup B$ and hence will get a final topology, whereas the intersection $A \cap B$ has inclusions $j_A : A \cap B \hookrightarrow A$ and $j_B : A \cap B \hookrightarrow B$ and therefore gets an initial topology. The set-difference is just another version of a subspace.

Definition 6.24 (Union, intersection, set-difference)**.** *Let (X, R) and (Y, S) be union compatible topological datatypes. Then we call*

$$(X, R) \underline{\cup} (Y, S) := (X \cup Y, R \cup S)$$

the topological union *and*

$$(X, R) \underline{\cap} (Y, S) := (X \cap Y, R^+ \cap S^+)$$

the topological intersection *and*

$$(X, R) \underline{\setminus} (Y, S) := (X \setminus Y, (X \setminus Y)^2 \cap R^+)$$

the topological difference *of (X, R) with (Y, S).* □

Remark 6.25. *We need to verify that the above concepts are well defined.*

Union. *The topological union is easily recognized as the final relation of the inclusions $X \hookrightarrow X \cup Y$ and $Y \hookrightarrow X \cup Y$.*

Intersection. *Let be $A = X \cap Y$. According to Lemma 6.6 $A^2 \cap R^+$ is initial for the inclusion $A \hookrightarrow X$ and $A^2 \cap S^+$ is initial for $A \hookrightarrow Y$. According to Theorem 6.4 the infimum of these relations is the left hand side of the equation*

$$(A^2 \cap R^+)^+ \cap (A^2 \cap S^+)^+ = R^+ \cap S^+$$

and it is an easy exercise to verify that this, indeed, is an equation.

Difference. *This is the subspace according to Lemma 6.6.*

Remark 6.26. *Note, that the topological union is a glueing if one of the inclusion mappings $(X, R)|_{X \cap Y} \hookrightarrow (Y, S)$ or $(Y, S)|_{X \cap Y} \hookrightarrow (X, R)$ is continuous.*

6.3. **On the necessity of computing the transitive closure.** As seen in the previous subsection, the use of the transitive closure of a relation is suggested in some cases. On the other hand, the generating relation of the product topology of two spaces involves only the two initial relations and the diagonal relations and no transitive closure operator (cf. Corollary 6.21). In this subsection, we will show that computing the transitive closure becomes indispensable in many important cases.

Theorem 6.27. *Let (X, R) be a topological datatype. If a relational algebra language $\mathfrak{L}$ can decide continuity in* **Alex***, then it can decide for any $(a, b) \in X^2$, whether or not (a, b) lies in the transitive closure R^+ of the relation R.*

Proof. Let (Y, S) be the topological datatype consisting of the set $Y := \{a, b\}$ and the relation $S := \{(a, b)\}$ on Y. Denote by $\mathbf{X}$ and $\mathbf{Y}$ the Alexandrov spaces corresponding to (X, R) and (Y, S). If the language $\mathfrak{L}$ can decide, if the inclusion map $\mathbf{Y} \to \mathbf{X}$ is continuous, then $\mathfrak{L}$ decides whether or not $S \subseteq R^*$, i.e. whether $(a, b) \in R^*$ or not. Then in $\mathfrak{L}$ the following query can be formulated:

$$(a = b \text{ and } (a, b) \in R) \text{ or } (a \neq b \text{ and } (a, b) \in R^*),$$

which is equivalent to $(a, b) \in R^+$. □

Theorem 6.28. *If a relational algebra $\mathfrak{L}$ can compute any initial topology in* **Alex***, then it is a relational algebra with transitive closure.*

This means $\mathfrak{L}$ can compute the transitive closure R^+ of any relation R on any set X.

Proof. Let R be an arbitrary binary relation in a domain $\mathcal{D}$, hence the schema if R is $[a : \mathcal{D}, b : \mathcal{D}]$. Then with

$$X := \beta_{id \leftarrow a} \pi_a(R) \cup \beta_{id \leftarrow b} \pi_b(R)$$

the pair (X, R) is a topological datatype. As the subspace topology is a particular instance of initial topology, we can make the following construction: Consider the relation $S := \Delta_{X^2} \otimes R$ on $X^2 \times X$ which we shall, by abuse of notation, consider a relation on X^3. Let T be any relation which generates the subspace topology of

$$A := \{(a, b, c) \in X^3 \mid a = c \text{ or } b = c\} \subseteq X^3.$$

The projection $\pi_3 \colon A \to X$ onto the third factor induces the relation P on X, given by

$$P := (\pi_3 \times \pi_3)(T)$$

Claim. $R^+ \setminus R^0 \subseteq P \subseteq R^*$.

The claim implies that if the language $\mathfrak{L}$ computes T, then it computes also $R^+ = (P \setminus R^0) \cup R$. We shall now prove the claim.

Left inclusion. Let $(a, b) \in R^+ \setminus R^0$, i.e. $(a, b) \in R^+$ and $a \neq b$. This means that the elements $(a, b, a), (a, b, b) \in A$ satisfy $(a, b, a) S (a, b, b)$. As T generates the subspace topology on A, this implies that $(a, b, a) T^+ (a, b, b)$. Now, let $n \in \mathbb{N}$ be minimal such that

$$(a, b, a) T^n (a, b, b). \tag{15}$$

As $a \neq b$, this n must be positive. If T is viewed as a graph with vertices in A, then (15) can be viewed as a path without backtracking from $p_0 = (x_0, y_0, z_0) = (a, b, a)$ along vertices $p_i = (x_i, y_i, z_i) \in A$ to $p_n = (x_n, y_n, z_n) = (a, b, b)$. As, however, all vertices p_i are taken from A, we must have

$$x_i = a, \quad y_i = b \quad z_i = a \text{ or } b$$

for all $i = 0, \ldots, n$. Thus, $n = 1$. But $(a, b, a) T (a, b, b)$ implies $(a, b) \in P$.

Right inclusion. Assume now $(a, b) \in P$. If $a = b$, then, obviously, $(a, b) \in R^*$. So, assume further $a \neq b$. By assumption, there exist $x_1, x_2, y_1, y_2 \in X$ such that

$$(x_1, x_2, a) T (y_1, y_2, b),$$

which implies

$$(x_1, x_2, a) S^n (y_1, y_2, b) \tag{16}$$

for some $n > 0$, as $A^2 \cap S^+$ and T both generate the subspace topology. But (16) means that $(a, b) \in R^*$. □

This expressive power to compute the transitive closure, however, is only needed for the general case. Alternatively, the dimension of the relational datatypes can be given a fixed upper limit d, in which case the transitive closure R^+ is equal to $\bigcup_{i=1}^{d} R^i$, which can be computed by every relational algebra.

7. Conclusions

From the statement that architecture is a finite partitioning of the Euclidian real space which has a refinement into a cw-complex immediately, we show how to construct a topology for architectural elements. The statement holds in our opinion not only for architecture, but in fact for every engineering discipline dealing with spatial artefacts. Also, we feel that any Computer Aided Engineering (CAE) software should be able to store and retrieve such topological information.

We propose a data model for this task after introducing the relevant concepts from topology, employing the natural language for describing the various aspects of topological information: categories and functors. With that proposed data model, we show that every topological information of finite sets can be stored. Therefore the data model is effective, a result which is interesting in the light of [5], where it has been found to be efficient, too, because no other data structure with this property can have a better asymptotical storage complexity. The complexity even becomes linear in practical CAE applications, because in these cases only "docile" topologies with linear complexity are likely to come up.

We show then, how to express topological constructions with relational database queries and, on the other hand, transform the basic query operators themselves into topological constructions. This becomes possible under the premise that the underlying query language can express relational algebra with transitive closure. Alternatively, the dimension of the modeled space can be given a fixed upper bound, thus avoiding the transitive closure for these special cases. In practical CAE for threedimensional artefacts this should hardly be a limitation.

To put it short, the proposed data model is a topological extension of the relational database model.

Acknowledgements

The authors thank Anita Höschele for some contribution. The anonymous reviewers are thanked for helpful remarks towards improving the exposition of the article. Prof. Dr. Niklaus Kohler is thanked for giving the authors the opportunity to work in this field.

Funding

This work is funded by the Deutsche Forschungsgemeinschaft (DFG) in the research project KO 1488/8-1, 8-2 "Architektonische Komplexe".

References

[1] Mäntylä, M. (1988) *An introduction to solid modeling*. Computer Science Press, Rockville, MD.

[2] Paul, N. and Bradley, P.E. (2003) Topological houses. In: *Proceedings 16th International Conference on the Applications of Computer Science and Mathematics in Architecture and Civil Engineering (ikm)*.

[3] Alexandrov, P. (1937) Diskrete Räume. *Matematiećeskij Sbornik*, **44**, 501–519.

[4] Paul, N. and Bradley, P.E. (2005) Relationale Datenbanken für die Topologie architektonischer Räume. In: *Forum Bauinformatik 2005. Junge Wissenschaftler forschen*.

[5] Paul, N. (2008) Topologische Datenbanken für Architektonische Räume. PhD thesis, Universität Karlsruhe, Germany.

[6] Paul, N. (2007) A Complex-Based Building Information System. In: *Proceedings of the 24th Conference on Education in Computer Aided Architectural Design in Europe*. Frankfurt am Main, 26-29 September 2007.

[7] Hatcher, A. (2002) *Algebraic topology*. Cambridge University Press, Cambridge, MA. Also available at http://www.math.cornell.edu/∼hatcher/

[8] Maier, D. (1983) *The theory of relational databases*. Pitman, London.

[9] Codd, E.F. (1990) *The relational model for database management*. Addison-Wesley, Reading, MA.